

تعلم أساسيات الأندرويد بالعربية باستخدام منصة Android Studio



بدءاً من الصفر وبتسلسل متناسق، تعلم بالعربية المبادئ الأساسية لبرمجة تطبيقات الأندرويد .

Learn Basics Android Programming by doing Apps starting from Scratch



Author : Abboudi Aliwi | 2016-2015

الفهرس

المقدمة

بسم الله الرحمن الرحيم والصلاة والسلام على أشرف الأنبياء والمرسلين,, سيدنا محمد وعلى آله وصحبه وسلم. وبعد ...

أضع بين إيديكم هذا العمل المتواضع, والذي يحتوي على أسياسيات برمجة تطبيقات الأندرويد ويحتاجه المبتدئين والمتقدمين في المجال, فيوجد الكثير من الأضافات لم تكن موجودة سابقاً فهي جديدة لذلك الأفضل التعرف عليها, كما إتمنى من كل شخص عندما يقرأ هذا الكتاب يدعو الله لي ولوالدي بالتوفيق والرحمة, فمهما بلغنا من عمل, ما أوتينا إلا قليلاً, فإذاً لا تبخل بنشر إي علم مفيد وصالح للآخرين, فنحن في زمن تتسابق فيه العلوم بشتى أنواعها, وما زلنا نحتكر العلوم لأنفسنا, فهل تسائلت يوماً إنك راحل ؟! .. راجع نفسك أخي الكريم سوف تجد الكثير من العلوم بداخلك والتي تستطيع التبرع بها للغير لعله يحسن إستخدامها فيكون لك الأجر بذلك والفضل من بعد الله, كما إن هذا الكتاب حاولت قدر الإمكان إن اجتهد بنفسي ليكون شرح وافي وسلس للجميع, كما إعتذر عن وجود عن إي إخطاء فلابد إن يكون هناك أخطاء ولو كانت صغيرة, لأسباب منها إن وجهات النظر تختلف من عدة جوانب وأشخاص, وكما إنه أول كتاب لي أكتبه بهذا الحجم, وعن نفسي أحياناً متسرع بإي عمل أقوم به أو بأغلبها, لذلك أخطاء لا انتبه لها, ومع ذلك أنا مؤمن باللذي أقوم به, لعل يوماً ما غداً أرى المطورين العرب أصبح لديهم تلك الأعمال التي تذكر أمام منافسيها من الغرب, الكثير يرى إن هذا حلم إن نصل لما وصلوا له, ولكن لو تسائلت هل لديهم عقول و لا يوجد لدينا ! بالتأكيد لأ, الفرق إنه تعلم منذ صغره كيف يقوم بالإستفادة من وقته ومهاراته, فهو يصنع ونحن نشجع ونستخدم,, قلتها وما زلت أقولها إن أغلبنا عناصر متفرجة, الكثير منا يتابع كرة القدم مثلاً, هل أحد لاحظ إن الدول التي تفوقت بكرة القدم تكن متفوقة بعدة مجالات آخرى !؟, هل لاحظتم لماذا لم يكن نصيب للولايات المتحدة او المملكة المتحدة نصيب كبير بكرة القدم ؟ .. الكون يحتاج لتفكير, ابدأ بأفكارك الصغيرة وانطلق بها, وانشرها للعالم فلو كانت تستحق سوف تصل بإرداتك او بدون, فيوجد الكثير من الأعمال التي لها شهرة عالية لم تشتهر إلا بعد وفاة أصحابها, فلا تكن إنسان " فارغ ", واستعن بالله بكل أمورك فما خاب من توكل عليه سبحانه, و أسئل الله لي ولكم التوفيق ودوام الصحة والعافية في الجسد والمال والولد.

الكاتب

عبدالقادر عليوي (Abboudi Aliwi) من مواليد مدينة دمشق, من عائلة ينحدر أصلها إلى مدينة الميادين في دير الزور-سوريا, ولدت في اليوم العالمي للعمل 1 آيار 1996.

أقوم بكتابة هذا الكتاب وأبلغ من العمر 19 عاماً, لا أقول عن نفسي إلا إنني عاشق لجميع المجالات في الحاسوب, وأحاول تعلم ما استطعت منها, وأحب قراءة العلوم الحديثة, وكتب تطوير الذات والثقة بالنفس والكتب الدينية المنوعة, ولدي طموح بتعلم اللغات والثقافات المختلفة.

للتواصل معي على الفيس بوك : https://fb.com/Abboudi.Aliwi

رجاءاً التواصل معي فقط للضرورة ولوجود اقتراح أو ملاحظة, ولإي استفسار أو مشكلة

تستطيع إضافتها في قروبنا على الفيس بوك :

https://fb.com/groups/Programming.Android.apps/

لكي تفيد وتستفيد من خبرات وأجابات الآخرين.

رجاءاً من كل شخص, صغيراً أو كبيراً, رجالاً أو إناثاً, الدعاء لي ولوالدي بالتوفيق والسداد, والمغفرة والرحمة, الدعاء بنية خالصة للّه لعلها تكون سبباً لنا بالنجاة والفوز بلقائه سبحانه.

تم الإنتهاء من إعداد هذا الكتاب بتاريخ 18-11-2015 مـ.

هل يجب تعلم اللغة الإنجليزية أو لغة الجافا لبرمجة التطبيقات :

الأسئلة هذه تكررت كثيراً - **هل يجب تعلم لغة الجافا أولاً لكي أستطيع برمجة تطبيقات الاندرويد ؟**

- هل يجب إن أتعلم اللغة الإنجليزية لكي أستطيع تعلم الجافا والبرمجة ؟

الجواب سوف يكون لا, ليست مهمة, لإنجاز التطبيقات البسيطة, ولكن كلما زادت معرفتك بلغة الجافا, كلما زادت الاحترافية في تطبيقاتك, وكذلك اللغة الإنجليزية فهناك كلمات عامة تتكرر معك في جميع اللغات البرمجية وهناك مصطلحات خاصة بالجافا وهناك مصطلحات يجب معرفتها سوف تتكرر معك في إنشاء وبرمجة التطبيقات الصغيرة, وهناك كذلك مصطلحات تحتاج معرفتها في بيئة الأندرويد ستوديو لكي تعرف كيف تتعامل معها.

واللغة الإنجليزية الأفضل لك تعلمها لكي تستطيع متابعة الشروحات والدورات الإنجليزية, لأنها هي حالياً اللغة الأم للعلوم الحديثة, لذلك لكي تتقن إي مجال من العلوم يجب عليك تعلمها لتستطيع إن تأخذ المعلومات من ألسنتهم, إي من المصدر و لغة الجافا بحر واسع, تعلم منه ما يخص أو ما تحتاج في بناء تطبيقات الأندرويد, وكذلك لغات البرمجة بشكل عام كل ما تعلمت منها أكثر أصبحت لك أكثر سهولة التعامل مع اللغات الأخرى ما دمت تملك تفكيراً برمجياً.

لذلك من الأفضل لك قبل أن تخضع للبدأ في هذا المجال, تذكر إنه يوجد العديد من المصطلحات الإنجليزية في برمجة تطبيقات الأندرويد ولكن يجب عليك إن تستخدمها لتعرف معناها تماماً لأن لها معاني أحياناً تختلف برمجياً عن معناها الحرفي بالإنجليزية.

و أنصحك بتحميل بعض من الكتب التي تحتوي على أساسيات الجافا OOP , فهي أساسيات تعلمها جيداً ومن ثم إنطلق في الدخول لعالم برمجة تطبيقات الاندرويد.

نصيحة يوجد الكثير ممن يدعى البرمجة عربياً, فقط يقوم بشراء الأكواد ووضع إعلانته فيها ويتم نشرها, وللأسف المتجر Google Play أصبح أكثر من سخيف بهذه التطبيقات المتكررة لذلك ابتعد عن رفع التطبيقات دون تعديل وتطوير يذكر, فقط أستعن بالأكواد منها لا مشكلة إما إعادة رفعها, فهكذا أنت تنفر الناس من استخدام بيئة الأندرويد أو التحميل من متجره. وأعتقد يوماً ما سوف تدرك الاندرويد هذه المشكلة وسوف تقوم بإزالة كل التطبيقات المتكررة.

نظام الأندرويد (Android - operating system)

ما يهمنا معرفته هو إن هذا النظام يعمل على الهواتف الذكية, وللعديد من الشركات المختلفة, كونه نظام مفتوح المصدر, بالإضافة إلى إنه مجاني, وفي عام 2005 م أصبح مُلك لشركة Google Inc.

بدأ العمل عليه في عام 2003 م. وتم إطلاق نسخه التجريبية في عام 2007 م, وحظى أول تجربة على جهاز من شركة HTC.


شعار نظام الأندرويد

والآن أواخر عام 2015 م, أشهر الشركات التي تستخدم نظام الأندرويد في أجهزتها هي :

Google Nexus, HTC, Samsung, LG, Sony, Motorola, Huawei, Lenovo,

وغيرها الكثير من الشركات الصينية والغير معروفة.

النظام يحتوي على إصدارات كثيرة بدأ مع النسخة **1.5 كب كيك** وإصداراته تبدأ اسمها مع الأحرف الهجائية الإنجليزية, وحتى الآن آخر إصدار هو **6.0 مارشميلو**.

إصدارات نظام الأندرويد : (Android History)

Android 1.0 Alpha
2008/9/23 م.



Android 1.1 Beta
2009/2/9 م.



Android 1.5 Cupcake
2009/4/27 م. ومن الإضافات دعم تصوير الفيديو للكاميرا + لوحة المفاتيح على الشاشة + النسخ واللصق من المتصفح



Android 1.6 Donut

2009/9/15 م. ومن الإضافات مربع البحث السريع + تحسين الكاميرا و خيارات لها بالإضافة إلى المعرض + يدعم قياس الشاشات المختلفة + متجر جوجل بلاي + اتصال vpn إلخ.



Android 2.1 Eclair

2009/10/26 م. ومن الاضافات دعم منصات مختلفة للبريد الإلكتروني وحسابات متعددة مع جوجل و الفلاش في الكاميرا + البحث داخل الرسائل + اللمس المتعدد بلوحة المفاتيح + المفضلة بالمتصفح



Android 2.2 Froyo

2010/5/20 م. ومن الاضافات دعم الفلاش بلاير تشغيل الفيديو بالمتصفح + دعم جهات الاتصال والبريد للنسخ الاحتياطي في خوادم جوجل + نقطة الوصول لمشاركة الواي فاي + تغيرات على المتصفح + الاتصال الصوتي بلوتوث



Android 2.3 Gingerbread

2010/12/6 م. ومن الإضافات توفير الطاقة + مؤشر يساعدك خيارات لتحديد ونسخ النص + نظام NFC + توافق عمل التطبيقات في الخلفية + دعم الكاميرا الأمامية لمكالمات الفيديو + مدير التنزيلات



Android 3.0 Honeycomb

2011/2/22 م. ومن الإضافات توافق التطبيقات المصغرة مع شاشات مختلفة + تطبيقات أساسية مثل بريد جوجل ويوتيوب + التبديل بين الأحرف والأرقام في لوحة المفاتيح + التصفح الخفي و علامات التبويب



Android 4.0 Ice Cream Sandwich

2011/10/18 م. ومن الإضافات سرعة وسلاسة أكبر بالمتصفح + نظام استخدام البيانات لتعيين حد + إزرار الخيارات بالأكشن بار + التعرف على بصمة الوجه لفتح القفل + توفير مساحة للتطبيقات



Android 4.1 Jelly Bean

2012/7/9 م. تطبيق جوجل الآن + مشاهدة الصور التي تم التقاطها الآن من خلال التمرير + الإشعارات تحتوي على معلومات أكثر + نتائج بحث جوجل تشمل حتى اجابات على الاسئلة +



Android 4.4 KitKat

2013/10/31 م. ومن الإضافات تحسين جوجل الآن + ترقية تطبيق Hangouts + وضع ملأ الشاشة للتطبيق اي اخفاء الازرار وشريط الحالة والاكشن بار + التحكم بشكل اكبر بتطبيقات التخزين مثل جوجل درايف لادارة الملفات المخزنة



Android 5.0 Lollipop

2014/11/12 م. ومن الإضافات وأهمها نظام التصميم الحديث الماتريال ديزاين + شاشة قفل جديد تظهر الاشعارات عليها + عمر اطول للبطارية مع توفير الطاقة + بحث في الإعدادات + إضافة زائر لجهازك ليتصفح جهازك بأمكانيات محدودة + تغيير الوان الشاشة + الوصول للتطبيقات الاخيرة حتى بعد اطفاء الجهاز + وصول اسرع للاعدادات واستخدام الذاكرة والبطارية إلخ



Android 6.0 Marshmallow

2015/10/05 م. ومن الإضافات حد الصلاحيات للتطبيق + دعم قارئ البصمة ليصبح متاح للمبرمجين التعامل معه + تقنية جديدة لإطالة عمر البطارية وحفظ الطاقة + نسخ احتياطي تلقائي + مراقبة اداء التطبيقات واستهلاكها للذاكرة

بنتظار الإصدار القادم .. والذي سوف يكون بالتأكيد كذلك اسم لحلوى ويبدأ بحرف الـ N .

برنامج أندرويد ستوديو : (Android Studio)

هو البرنامج الذي سوف نستخدمه لبرمجة تطبيقات الأندرويد, سابقاً كنا نستخدم برنامج **الأكليبس** **Eclipse** وما زال إلى الآن البعض يستخدمه, مع العلم إنه تم إعلان إيقاف تطوير الأدوات ADT الخاصة بنظام اندرويد للأكليبس في **نهاية عام 2015 م**.


ايقونة برنامج اندرويد ستوديو

وذلك لكي تعطي الشركة قدر الإمكان الوقت والجهد والتفكير على جعل الأندرويد ستوديو أكثر تقدم واحترافية لهذا المجال فقط, بينما الأكليبس تستطيع أستخدامه في لغات وأعمال برمجية آخرى.

تم إطلاقه في Google I/0 لعام 2013 م. كما إنه إلى الآن ما زال يتلقى العديد من التحديثات والتطويرات.

حسب الشرح المستخدم في هذا الكتاب, إصدار برنامج أندرويد ستوديو هو Android Studio 1.4

مدير إضافات الأندرويد (SDK Manager)

اختصاراً لـ **S**oftware **D**evelopment **K**it هي الأدوات والحزم والإضافات التي توفرها أندرويد للمطورين, وتستفيد منها في حال هناك إصدار جديد لنظام الأندرويد, تقوم بعمل تحديثات من أجل برمجة التطبيق لكي يدعم آخر الإصدارات فهي مهمة ويتم ربطها تلقائياً مع الأندرويد ستوديو. دائماً أجعلها محدثة لآخر إصدار كما يمكنك إن تقوم بإعداد داخل الإندرويد ستوديو لتنبيهك لوجود إي تحديث جديد.

لفتح نافذة الـ SDK :

قم بتشغيل برنامج الأندرويد ستوديو ومن شاشة الترحيب أختر Configure ثم Settings سوف يتم فتح نافذة جديدة بإعدادات البرنامج, من الجهة اليسرى أختر

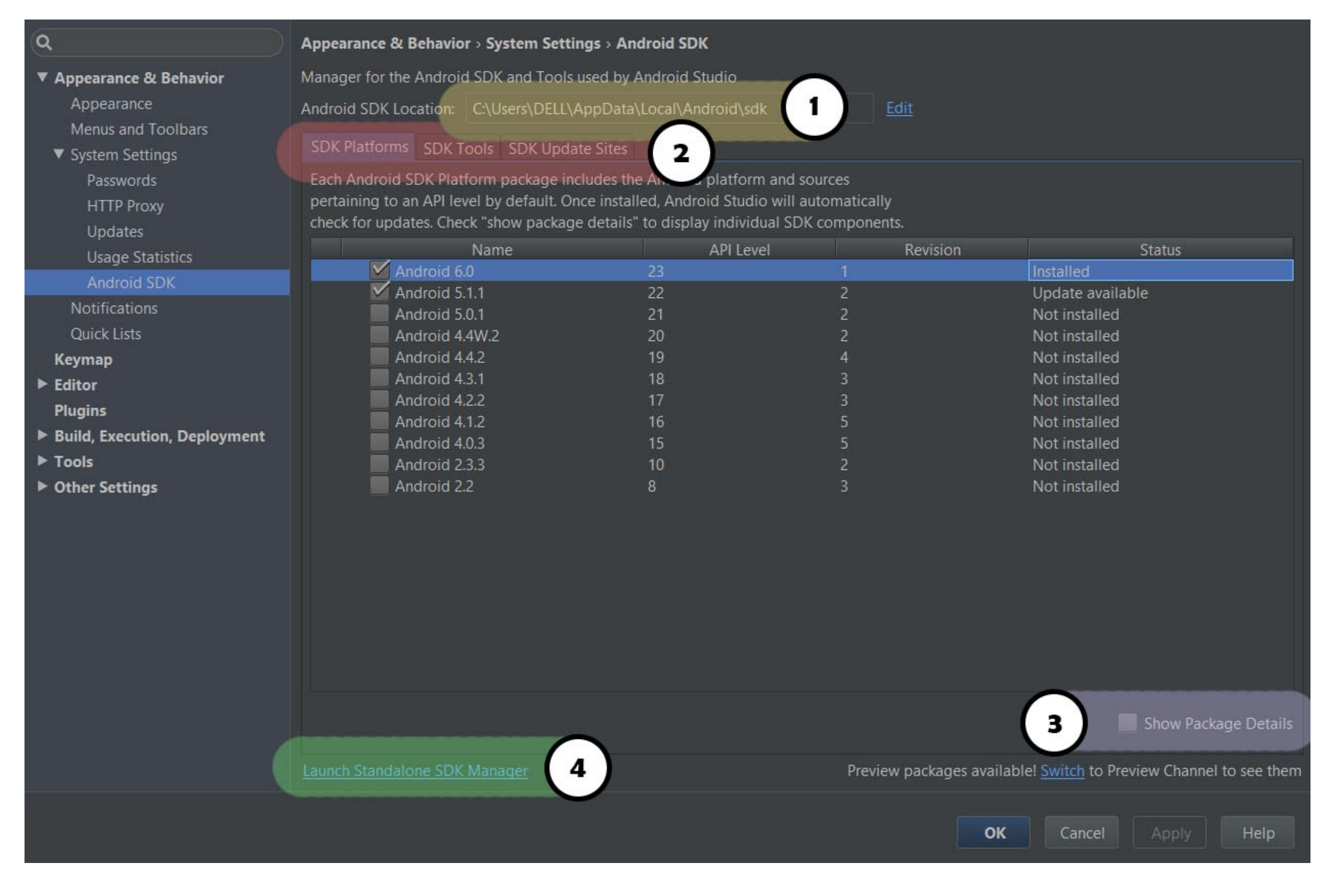
Appearance & Behavior > System Settings > Android SDK

أو تستطيع فتحها إذا كنت تعمل على أحد المشاريع, من شريط القائمة العلوي, إضغط على الأيقونة الخاصة بـ SDK



أو من الشريط العلوي

Tools > Android > SDK Manager



يوجد لدى البعض من الأخوة لديهم مشكلة في تنزيل SDK منفصلة عن البرنامج, أو إستخدام SDK سابقة في الجهاز يريدوا إستخدامها, لذلك تأكد دائماً من مسار الـ SDK في الصورة السابقة **رقم 1**.

حسب الصورة السابقة نقطة رقم 2 :

دائماً تأكد من إنك أخترت أعلى إصدار ليتوافق جهازك مع إي جهاز يعمل على نظام اندرويد, بالإضافة إلى إنه ومن الضروري إن يحتوي على إصدار واحد على الأقل لتستطيع البدء في إنشاء المشاريع.	**SDK Platforms**
إضافات ليدعمها الأندرويد ستوديو واستخدامها في مشاريعك, سوف يتم شرح المهم منها في نقاط قادمة, تأكد إن لديك الإصدار الأخير إي محدثة دائماً.	**SDK Tools**
يوجد بعض الخدمات التي تقدمها المواقع او الشركات والتي تعطيك رابط يحتوي على المنصات الخاصة بها SDK تستطيع إضافة الرابط هنا, وسوف تظهر إضافته في الأقسام الأخرى إذا كان يحتوي.	**SDK Update Sites**

بالنسبة للنقطة رقم 3 :

فهي لتظهر لك تفاصيل أكثر, مثلاً بعد الضغط عليها سوف تظهر لك الإصدار بكامل الحزم التي داخله وسوف أشرح في نقاط قادمة أهم ما يجب تحميله من كل إصدار.

وأما بالنسبة للنقطة 4 :

فهي لفتح نافذة الـ SDK بشكل مستقل, وهي النافذة التي تعودنا عليها في الإصدارات السابقة و ما زالت إلى الآن موجودة في إكليبس, وإندرويد ستوديو ولكن احتمال كبير يتم إلغاء فتحها بشكل مستقل, لأنها أصبحت أكثر سهولة كما ظهرت لنا بالصورة السابقة.

سوف يتم إعطاء لمحة سريعة عن نافذة SDK المستقلة وما أهم ما يجب تحميله منها, لمراجعة سابقة في هذا الأمر : http://andrody.com/?p=70

نافذة (Standalone SDK Manager)

ما قد تم تحميله في الصورة السابقة, سوف يتم شرحه سواء كان للـ SDK الحديثة او المستقلة, وهو الذي يهمنا من أجل البدء ببرمجة المشاريع الخاصة بنا, ولكن أحرص على أن تكون دائماً أحدث الإصدارات لديك منها.

// Android SDK Tools يتم تحديثها كل فترة, فهي تأتي بإصلاح أو تحديث أو تطوير بعض الأمور بالبرنامج, مثلاً مشكلة في استايل معين, مشكلة في المحاكي الافتراضي إلخ. لذلك يفضل تبقى متأكد إنك تستخدم أعلى إصدار دائماً.

// Platform, Build-tools تأكد إن Rev. يتوافق مع الـ API الأخير لديك لكي تستطيع العمل بدون مشاكل,تحديداً Build-tools سوف أشير إليها عندما نبدأ البرمجة بمشروعنا الأول.

// SDK Platforms كما ذكرت سابقاً يجب إن يحتوي على الأقل واحدة منها ويفضل أحدث إصدار لكي يعمل تطبيقك على أحدث الأجهزة.

داخل كل حزمة سوف تجد **System Image** وهي اختيارية في حال أردت إعداد وتركيب جهاز افتراضي محاكي تقوم بتجربة تطبيقاتك عليه قبل نشرها. (سوف نتكلم عن المحاكي في نقاط قادمة).

Android Support Library // مهمة لدعم إضافات كثيرة منها الـ fragments كذلك تحتاجها في حال أردت معاينة اللياوت التي تعمل عليها على إصدار سابقة وبكفاءة.

Android Support Repository // مطلوبة لأنها تحتوي على العديد من مكاتب الدعم والتي لا غنى عنها في برمجة إي مشروع.

Google Play services // لدعم خدمات جوجل بلاي مع أمثلتها, مثل اعلانات جوجل وسحابة جوجل واحصائيات جوجل وغيرها.

Google Repository // يحتوي على مكاتب الدعم الخاصة بجوجل.

المستويات في النظام (API Level)

ما يهم معرفته هو المستوى لكل إصدار, أي البرمجة في نظام الأندرويد على مستويات, مثلاً الـ API للأصدار الأخير مارشميلو هو 23. والـ API لإصدار الكت كات هو 19. إذا أثناء البرمجة إنت تحدد أقل إصدار يعمل عليه تطبيقك + أعلى إصدار ومن الطبيعي جداً أعلى إصدار هو آخر إصدار لديك في حزمة الـ SDK. لكي تضمن إن تطبيقك سوف يعمل على الإصدارات الحديثة لنظام الأندرويد. وكلما أصبح أقل الـ API زادت عدد الملفات التي يأخذها تطبيقك لكي يعمل على جميع الإصدارات . ولكن هكذا سوف تمنع بعض المميزات التي لا تقبل إلا الإصدارات الحديثة, فالأفضل لك إن تختار ليفل او مستوى موافق للجوالات الأكثر انتشار حالياً. بينما التطبيقات الضخمة بالغالب تستخدم جميع المستويات ولكن لتفعيل المميزات تضع شرط أثناء البرمجة إذا كان الجوال يعمل على ليفل كذا وكذا إذا .. وهكذا .. من أجل أن يكون تطبيقهم متوافق لجميع الأجهزة. لا تقلق بقراءة هذه المعلومة فقط للمعرفة .. أثناء البرمجة سوف تتعرف عليها بشكل أدق وفي هذا الكتاب سوف نبين ما المقصود بقولنا هذا من خلال إنشائنا أول تطبيق.

فلا شك عند إستخدامك إي جهاز اندرويد, بعد فتح الضبط – حول الجهاز – إصدار اندرويد. نظام أندرويد كما ذكرت فهو يحتوي على عدة اصدارات مختلفة.

أثناء برمجة مشروعك, سوف تعطيه أقل إصدار يعمل عليه تطبيقك, وأعلى أصدار حسب الموجود لديك بـ SDK .

لو جهازك الاندرويد نظامه مثلاً Android 3.0 هذا يعني إنه إصدار قرص العسل هونيكومب ويحمل API Level = 11.

وأنت قمت ببرمجة تطبيق حديث أقل ليفل 16 وأعلى ليفل 23 . إذا هذا التطبيق لن يعمل على جهازك. لذلك مهم لك هذا الجدول للتعرف على الإصدارات والمستويات لها :

الإصدار	المستوى
Android 1.0	1
Android 1.1	2
Android 1.5 Cupcake	3
Android 1.6 Donut	4
Android 2.0 Eclair	5
Android 2.0.1 Eclair	6
Android 2.1 Eclair	7
Android 2.2–2.2.3 Froyo	8
Android 2.3–2.3.2 Gingerbread	9
Android 2.3.3–2.3.7 Gingerbread	10
Android 3.0 Honeycomb	11
Android 3.1 Honeycomb	12
Android 3.2 Honeycomb	13
Android 4.0–4.0.2 Ice Cream Sandwich	14
Android 4.0.3–4.0.4 Ice Cream Sandwich	15
Android 4.1 Jelly Bean	16
Android 4.2 Jelly Bean	17
Android 4.3 Jelly Bean	18
Android 4.4 KitKat	19
Android 4.4 KitKat with wearable extensions	20
Android 5.0–5.0.2 Lollipop	21
Android 5.1–5.1.1 Lollipop	22
Android 6.0 Marshmallow	23

التحميلات المطلوبة للبدء ببرمجة تطبيقات الأندرويد

ما يتطلب تحميله لكي تبدأ برمجة تطبيقات الأندرويد, منصة الجافا لجهازك بالإضافة إلى برنامج الأندرويد ستوديو والذي يأتي معه حزمة الـ SDK.

ما يتطلبه جهازك ليتم تشغيل برنامج أندرويد ستوديو بكفاءة :

لأجهزة نظام الويندوز //

- نظام مايكروسوفت ويندوز فيستا أو أحدث للنواتين 64 و 32 بت
- الرام كحد أدنى 2 ويفضل 4 ما فوق
- 400 ميغا فما فوق فارغ من مساحة الهارد ديسك
- ما يتجاوز 1 جيجا من أجل حزمة الـ SDK
- كحد أدنى دقة الشاشة 1280×800
- أدوات تطوير الجافا JDK 7 فما فوق



لأجهزة نظام ماك أو إس //

- ماك أو إس 10.8.5 أو أعلى
- الرام كحد أدنى 2 جيجا ويفضل 4
- كذلك 1 جيجا مساحة من أجل حزمة الـ SDK
- دقة الشاشة كحد أدنى 1280×800
- بيئة جافا JRE 6 فما فوق
- أدوات تطوير الجافا JDK 7 فما فوق



لأجهزة نظام اللينكس //

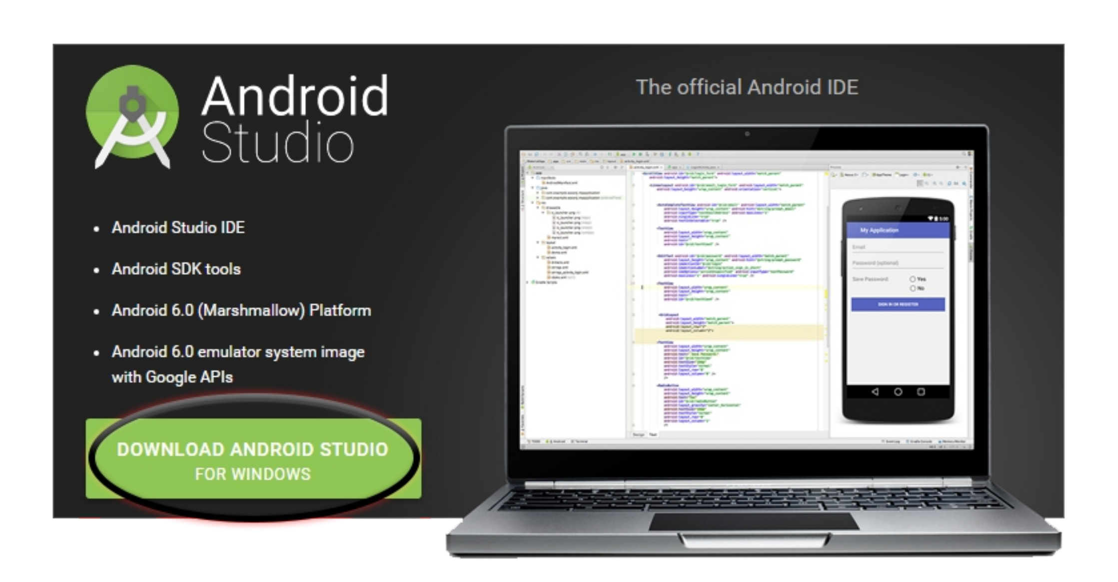
- ليكنس جينومي أو كدي
- مكتبة GNU C 2.15 أو أعلى
- رام 2 جيجا ويفضل 4
- 400 مساحة فارغة من القرص
- كذلك 1 جيجا مساحة من أجل حزمة الـ SDK
- دقة الشاشة كذلك كحد أدنى 1280×800
- أوراكل أدوات تطوير الجافا JDK 7 فما فوق.



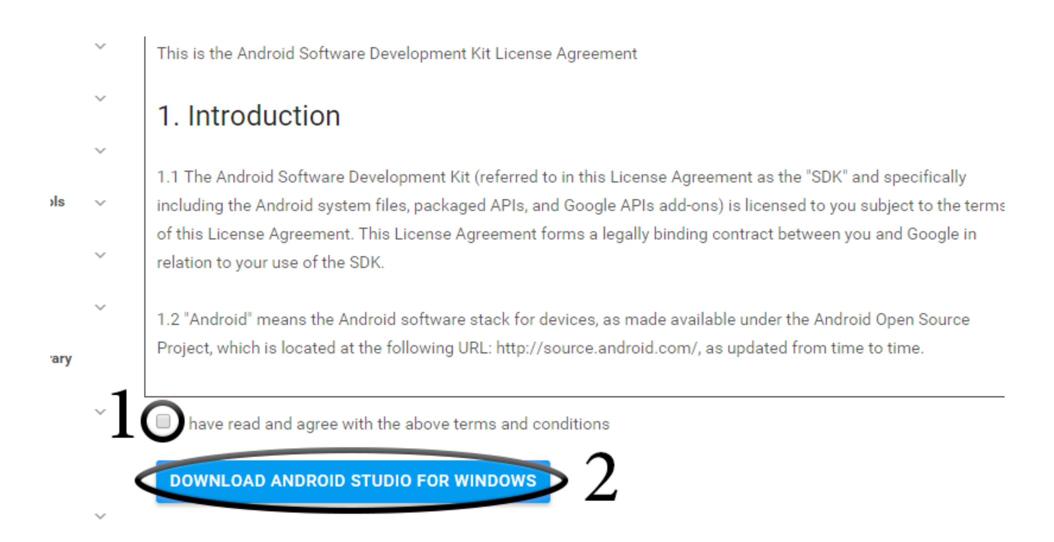
لتحميل الأندرويد ستوديو, افتح محرك البحث **Google** واكتب في شريط البحث : **Android SDK**

أو اتبع الرابط التالي : http://developer.android.com/sdk/

- بعد فتح الرابط إضغط على زر DOWNLOAD ANDROID STUDIO



1- ثم أضغط على (التشيك بوكس) للموافقة على الشروط والأحكم 2 بعدها سوف يصبح لديك زر التحميل باللون الأزرق اضغط عليه لبدء التنزيل (DOWNLOAD ANDROID STUDIO)



الآن سوف يبدء بالتنزيل, ننتظره حتى الأنتهاء ..

++ للدول المحجوب عنها موقع مطوري جوجل, تستطيع تحميل النسخة 1.4 :

الجزء الأول 572 ميغابايت / http://goo.gl/2cXkFa

الجزء الثاني 551 ميغابايت / http://goo.gl/bP8Koc

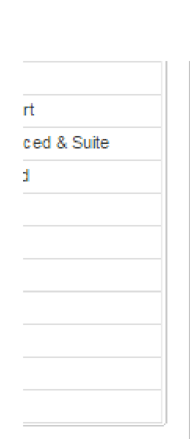
كلمة سر فك الضغط : andrody.com

الآن يجب علينا تحميل أدوات مطوري الجافا لجهازنا (**JDK**) نتبع الرابط التالي: http://goo.gl/Lp9A6i

- نضغط على زر التنزيل بجانب JDK



- ثم الضغط على قبول اتفاقية الترخيص



Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- Java Developer Newsletter: From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK 8u65 Checksum
JDK 8u66 Checksum

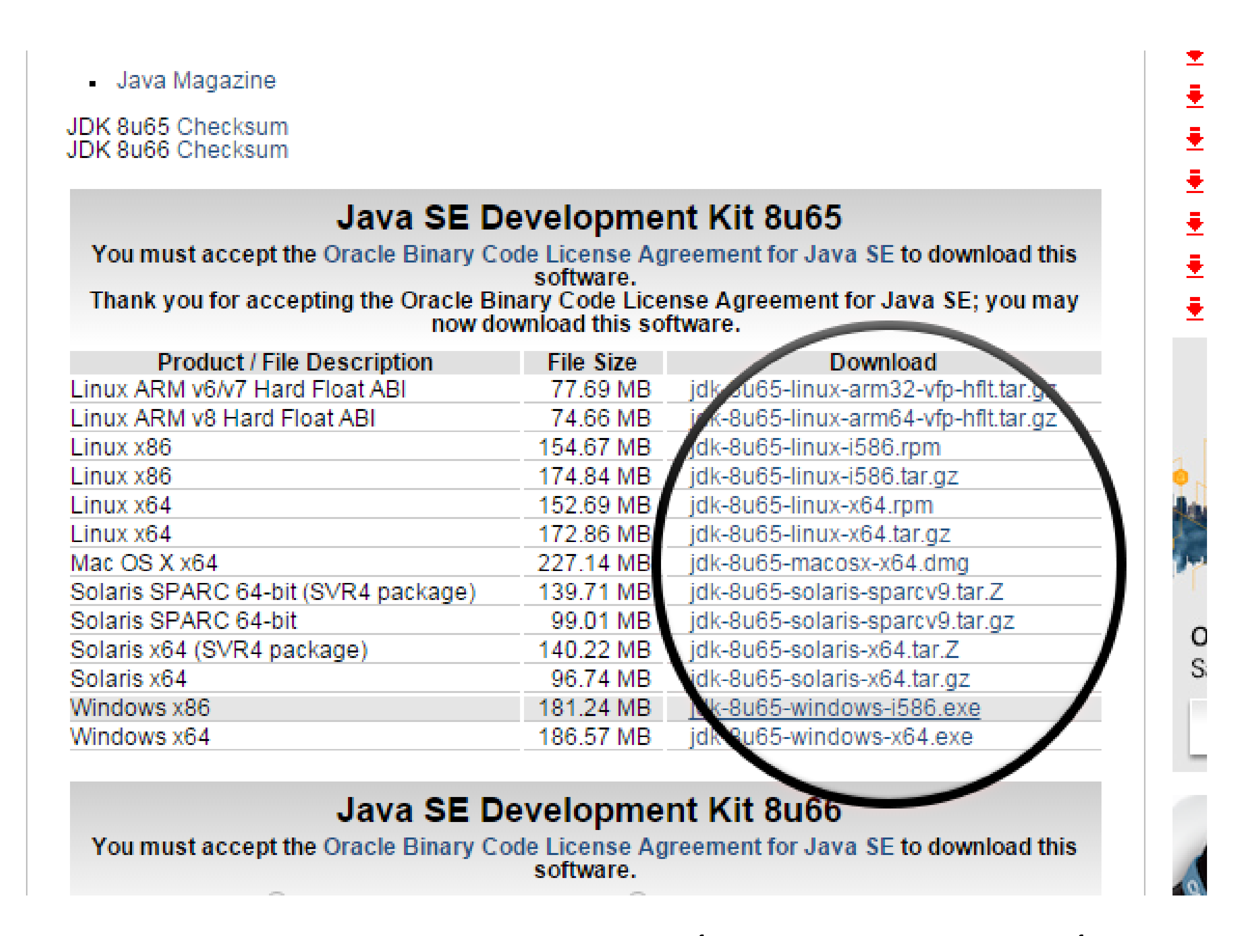
Java SE Development Kit 8u65

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Accept License Agreement　　Decline License Agreement

Product / File Description	File Size	Download
Linux ARM v6/v7 Hard Float ABI	77.69 MB	jdk-8u65-linux-arm32-vfp-hflt.tar.gz
Linux ARM v8 Hard Float ABI	74.66 MB	jdk-8u65-linux-arm64-vfp-hflt.tar.gz
Linux x86	154.67 MB	jdk-8u65-linux-i586.rpm
Linux x86	174.84 MB	jdk-8u65-linux-i586.tar.gz
Linux x64	152.69 MB	jdk-8u65-linux-x64.rpm
Linux x64	172.86 MB	jdk-8u65-linux-x64.tar.gz

- ثم نضغط على رابط التنزيل بجانب النسخة التي توافق نظام جهازنا



- Java Magazine

JDK 8u65 Checksum
JDK 8u66 Checksum

Java SE Development Kit 8u65

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux ARM v6/v7 Hard Float ABI	77.69 MB	jdk-8u65-linux-arm32-vfp-hflt.tar.gz
Linux ARM v8 Hard Float ABI	74.66 MB	jdk-8u65-linux-arm64-vfp-hflt.tar.gz
Linux x86	154.67 MB	jdk-8u65-linux-i586.rpm
Linux x86	174.84 MB	jdk-8u65-linux-i586.tar.gz
Linux x64	152.69 MB	jdk-8u65-linux-x64.rpm
Linux x64	172.86 MB	jdk-8u65-linux-x64.tar.gz
Mac OS X x64	227.14 MB	jdk-8u65-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.71 MB	jdk-8u65-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.01 MB	jdk-8u65-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	140.22 MB	jdk-8u65-solaris-x64.tar.Z
Solaris x64	96.74 MB	jdk-8u65-solaris-x64.tar.gz
Windows x86	181.24 MB	jdk-8u65-windows-i586.exe
Windows x64	186.57 MB	jdk-8u65-windows-x64.exe

Java SE Development Kit 8u66

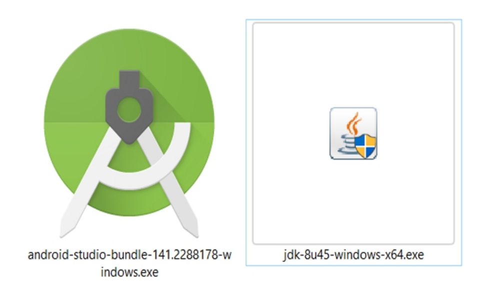
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

سوف يبدأ بالتنزيل, ولمن يواجه صعوبة أو حظر للموقع تستطيع تحميلها بروابط خارجية من هنا :

- لأجهزة نظام ويندوز 64 بت / http://goo.gl/iNmqaG

- لأجهزة نظام ويندوز 32 بت / http://goo.gl/Znr1xm

هكذا نكون أنتهينا من التنزيلات المطلوبة . . والناتج النهائي لدينا الآن :



والآن قم بتثبيتها بكل سهولة, فقط فتحها والضغط على زر التالي .. حتى الانتهاء. وقم أولا بتحميل

أدوات مطوري الجافا JDK من ثم **برنامج الأندرويد ستوديو**.

الخطوات الأولى التي يجب عليك القيام بها

في البداية وقبل تشغيل برنامج أندرويد ستوديو, قم بالتالي, **تعيين وصول جهازك للجافا**, بالخطوات التالية لنظام الويندوز (غير ضرورية فقط لمن يواجه مشكلة في حال لم يتم تشغيل أندرويد ستوديو)

– اضغط بالزر الأيمن على جهاز الكمبيوتر واختر خصائص, ثم من النافذة التي على اليسار اختر Advanced system setting.

– سوف تظهر لك نافذة صغيرة, من ضمنها خيار في الأعلى اسمه Advanced, من داخله يوجد اسمه Environment Variables اضغط عليه, سوف تخرج لك نافذة جديدة كذلك.

– النافذة الجديدة قسمين, في الأعلى User variables for وفي الأسفل System Variables, في الجزء السفلي اضغط على زر New, سوف تخرج لك نافذة جديدة.

– نافذة صغيرة مكونة من خانتين Variable name و Variable value, في داخل الاسم ضع : JAVA_HOME وأما في داخل القيمة ضع : – مسار ملف الجافا لديك في جهازك – مثال : C:\Program Files\Java\jdk1.8.0_40 مع تغيير رقم الاصدار المتوفر لديك.

– اضغط OK وتم الحفظ. بعدها جرب تشغيل البرنامج إذا اشتغل بشكل سليم مبروك عليك, وإذا لم يعمل, ارجع لنفس المكان. ونفذ التالي :

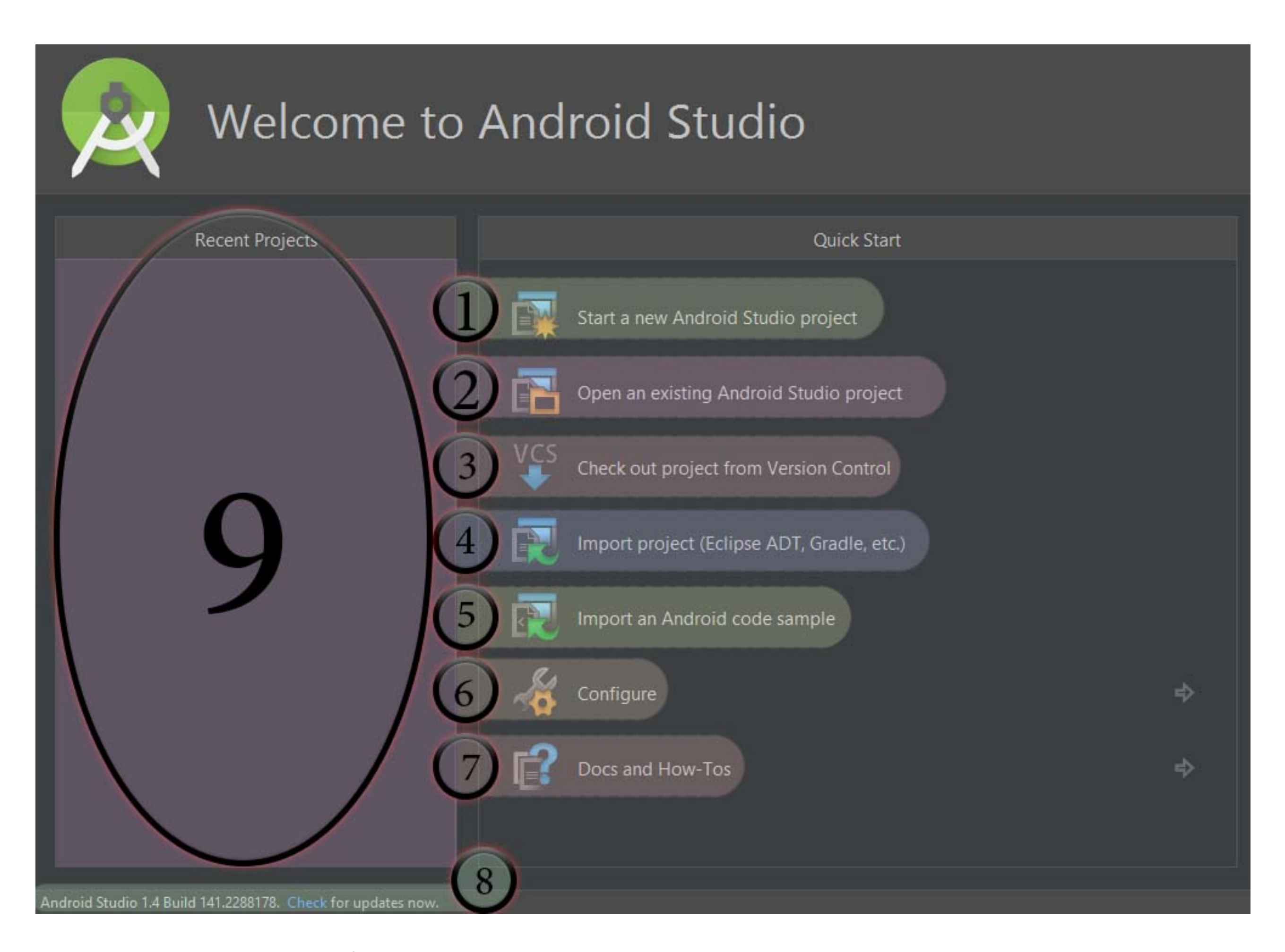
– في الجزء العلوي User variables for , اضغط على خانة PATH واضغط تعديل, او بشكل متكرر عليها لتظهر قائمة التعديل .. في خانة القيم تجد سطور موجودة, اذهب آخرها وضع الفاصلة المنقوطة ; ومن ثم نفس المسار الذي وضعته في خانة JAVA_HOME ولكن بإضافة \bin مثال : C:\Program Files\Java\jdk1.8.0_40\bin

– اضغط OK . في حال لم يعمل اندرويد ستوديو جرب تشغيله كمسؤول.

و الآن **" انتبه ! "** قبل أن تقوم بفتح برنامج الأندرويد ستوديو يجب عليك التوجه لحزمة الـ SKD وتحميل المطلوب منك كما ذكرنا سابقاً في موضوع || مدير إضافات الأندرويد : (SDK Manager) ||

بعد قيامك بتنزيل الحزم من الـ SDK, مع العلم يوجد نسخ من الأندرويد ستوديو تأتي معها الحزم ولكن تأكد بنفسك أولاً , تستطيع بعدها الأنتقال للمواضيع التالية. إذا واجهتك إي مشكلة لا تتردد بالتواصل معنا .

نافذة الأندرويد ستوديو الترحيبية



هذه النافذة الموضحة بالصورة, هي النافذة التي تظهر لك بداية تشغيل الأندرويد ستوديو, في حال لا يوجد مشاريع مفتوحة, وإذا كان يوجد مشروع تم فتحه مسبقاً يجب إغلاقه لتظهر لك هذه النافذة.

عند النقطة (1) // البدأ بإنشاء مشروع جديد.

و عند النقطة (2) // فتح مشروع موجود مسبقاً, وقد تم بنائه على الأندرويد ستوديو.

وعند النقطة (3) // جلب مشاريع من عدة مواقع, كل منها له طريقة معينة في الأستخدام لذلك سوف يترك شرحه لتدوينات لاحقة.

وعند النقطة (4) // استيراد مشاريع اندرويد المبنية على برنامج اكليبس, سوف يتم شرحه في موضوع مستقل.

وعند النقطة (5) // استيراد نماذج أكواد للأندرويد بسيطة أو قصيرة جاهزة.

وعند النقطة (6) // مرحلة الإعداد, الـ SDK والإضافات والإعدادات إلخ.

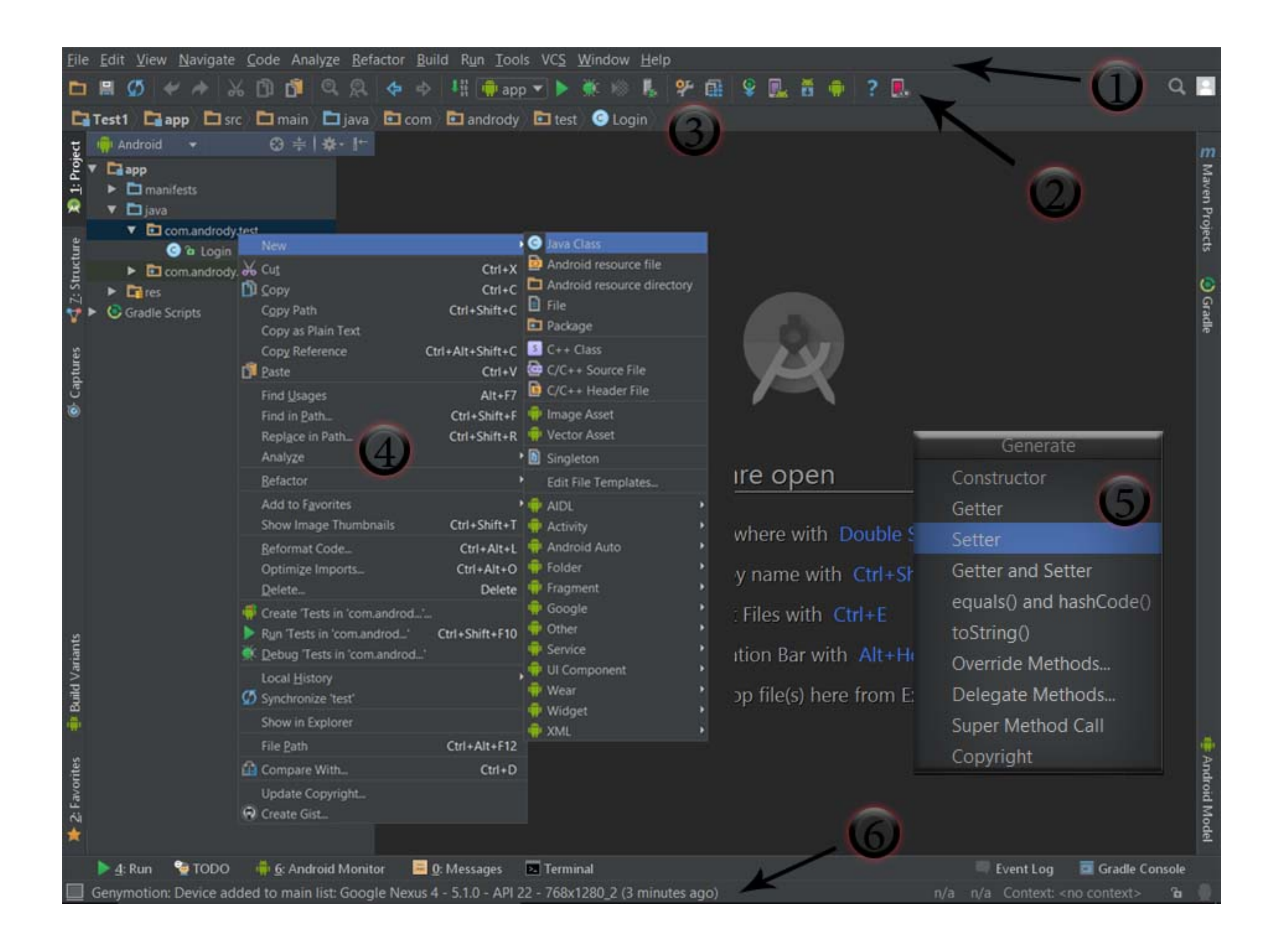
وعند النقطة (7) // تعليمات وشروحات لكيفية استخدام بيئة IntelliJ, إنتيليج إيديا وهي التي يعمل عليها برنامج أندرويد ستوديو.

وعند النقطة (8) // الإصدار الحالي لديك, و للتحقق من وجود إي تحديثات للبرنامج.

وعند النقطة (9) // المشاريع التي تم فتحها مؤخراً على برنامج الأندرويد ستوديو.

نافذة الأندرويد ستوديو

هذه النافذة الرئيسية لمشروعك على برنامج الأندرويد ستوديو, سوف نتعرف عليها بشكل سريع لكي تستطيع استخدام البرنامج بكل سلاسة.



حسب الأرقام الموضحة في الصورة السابقة, سوف أقوم بشرحها :

1 // القائمة الرئيسية (Main menu) :

القائمة الرئيسية للبرنامج, وتحتوي على عدة أوامر مختلفة, منها فتح وحفظ وإنشاء المشاريع, والإعدادات, وتنظيف المشروع وإعادة بنائه, وتصديره لملف .apk والأدوات وتحديث البرنامج وغيرها الكثير..

2 // شريط الأدوات (Main toolbar) :

شريط الأدوات الرئيسي, ويحتوي على الأزرار الأكثر استخداماً, وهي مهمة جداً. منها حفظ المشروع, مع العلم إن الأندرويد ستوديو يحفظ الملفات بشكل تلقائي, و امر النسخ واللصق وتشغيل المشروع وبناء التطبيق وفتح الـ SDK إلخ

3 // شريط التصفح (Navigation bar) :

شريط التصفح أو التنقل للوصول السريع لملفات المشروع والتنقل بينها لتحريرها.

4 // قائمة الخيارات (Context menus) :

قائمة تظهر بعد الضغط بالزر الأيمن للفأرة وتختلف حسب الموقع الذي ضغطت عليه

5 // القائمة المنبثقة (Pop-up menus) :

قائمة تستطيع إظهارها بالضغط على زر Alt + Insert أكثر من رائعة تفيدك مثلاً لاضافة أكواد او دوال حسب الملف المفتوح, كذلك تستطيع ادراج حقوقك على الملف المفتوح حسب الموجود لديك في اعدادات البرنامج.

6 // شريط الحالة (Status bar) :

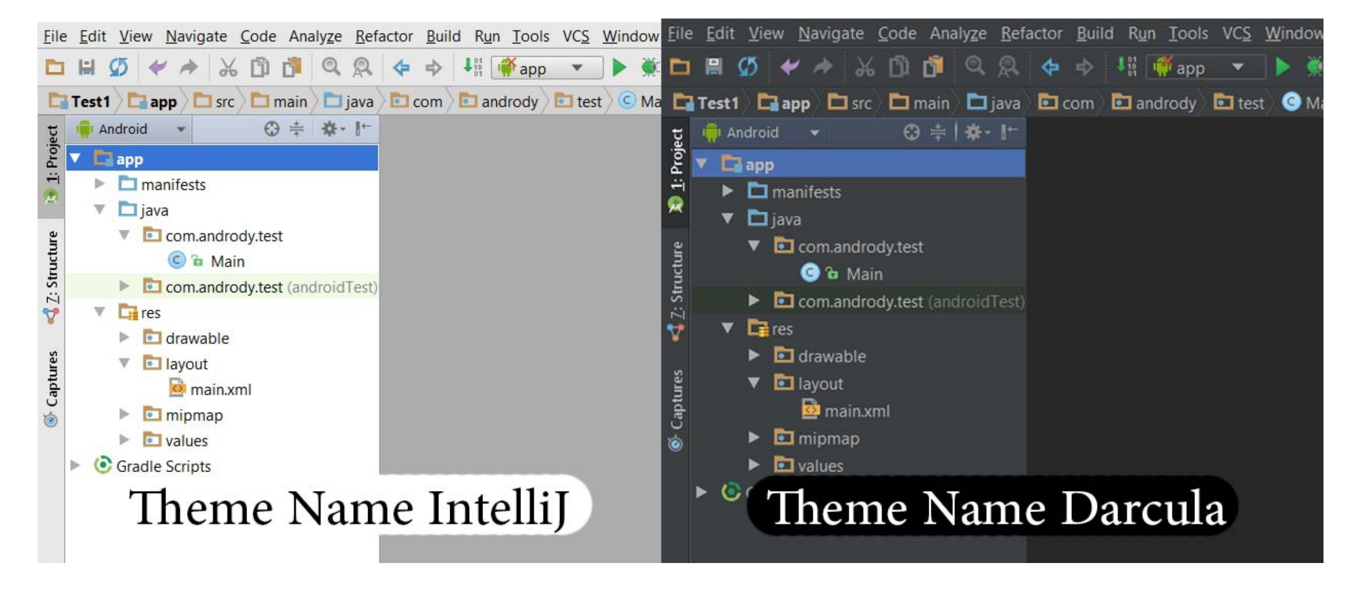
شريط يظهر لك مدة تنفيذ عملية, او تفاصيل أكثر لإي زر تؤشر عليه موجود في شريط الأدوات.

في شريط الحالة كذلك, تستطيع تغيير ترميز الملف الحالي, وتقفيل الملف من التعديل إي جعله للقراءة فقط بشكل مؤقت, كم يوجد رقمين تفصل بينهم نقطتان رأسيتان مثال 1:2 رقم واحد للدلالة على إنك في السطر الأول, والرقم 2 للدلالة على إنك عند الخانة أو الحرف الثاني من السطر الاول.

التعامل مع الأندرويد ستوديو

في هذا الموضوع إن شاء الله سوف نتناول طرق وخصائص وإضافات تسهل علينا بعض العمليات التي نريد القيام بها.

تغيير الثيم أو الاستايل الخاص ببرنامج الأندرويد ستوديو :



يوجد لديك ثيمات أساسية لأندرويد ستوديو وهي IntelliJ و Darcula, أستخدم منها ما يناسبك.

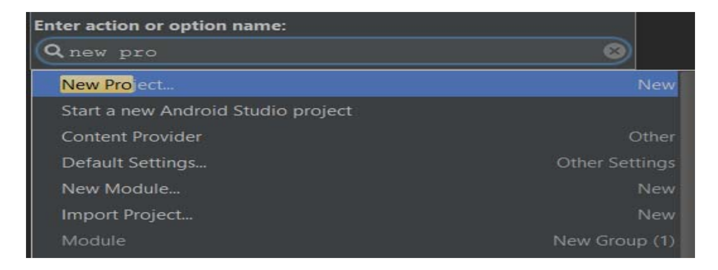
لتغيير الثيم توجه إلى إعدادات البرنامج بالضغط على Ctrl + Alt + S أو: File => Settings

سوف تظهر لك نافذة الإعدادات, من الجهة اليسرى إضغط على : Appearance & Behavior ثم Appearance ثم UI Options / Theme.

ملاحظة : تستطيع تغيير كل شيء يدوياً, حجم الخط, ألوان المحرر, نوع الخط, اختصارات المفاتيح, ترتيب الأدوات إلخ.

المساعد :

هل تفكر بالقيام بأمراً ما, ولكن لا تعلم إين تجده ؟! .. بإمكان المساعد مساعدتك بإي أمر تريد القيام به سواء كان أمر أو خيار. فقط قم بالضغط على Ctrl + Shift + A , وهذه صورة للتوضيح :



مشكلة الأحرف الغير مفهومة داخل الأندرويد ستوديو :

مشكلة الأحرف الغير مفهومة تتعلق بالترميز, لذلك تأكد دائماً من ترميز المشروع إنه UTF-8 وكذلك من الملف الحالي إنه نفس الترميز.

File → Settings → File Encoding :

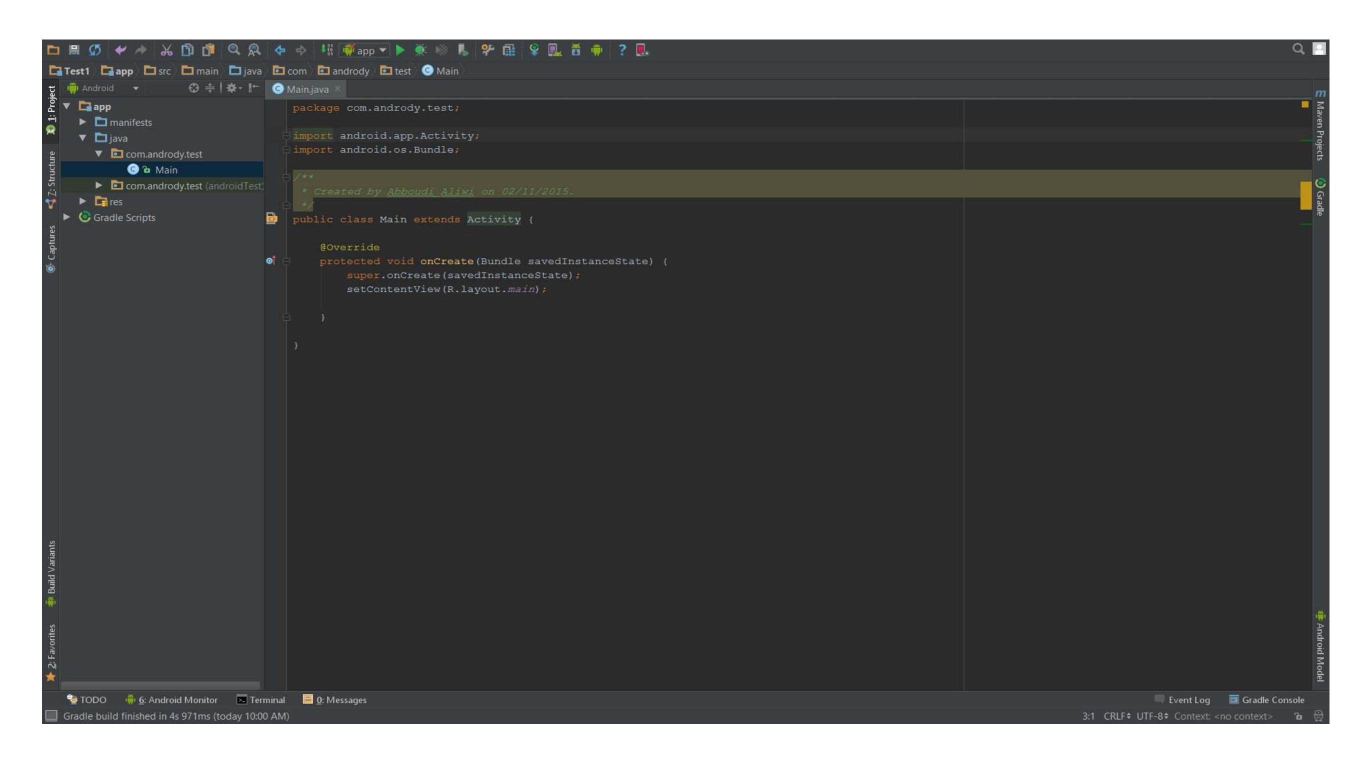
Project Encoding = UTF-8

IDE Encoding = UTF-8

والملف الحالي من الزاوية السفلى يساراً, يوجد الترميز تأكد إنه UTF-8.

وضعيات التركيز على مشروعك :

1. full screen mode :



لجعل برنامج الأندرويد ستوديو ملأ الشاشة للتركيز أكثر على البرنامج, من القائمة اختر View ثم Enter Full Screen و للخروج منه كذلك اختر Exit Full Screen.

2. presentation mode :

```java
package com.andrody.test;

import android.app.Activity;
import android.os.Bundle;

/**
 * Created by Abboudi_Aliwi on 02/11/2015.
 */
public class Main extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

    }

}
```

إما هذه الوضعية فالتركيز تماماً على الكود الذي تم فتحه, وهي مهمة لمن يكتب سطور برمجية كثيرة في اكتفتي واحد, لتفعيل هذا الوضع من القائمة اختر View ثم Enter Presentation Mode و للخروج منه كذلك اختر Exit Presentation Mode.

```java
package com.andrody.test;

import android.app.Activity;
import android.os.Bundle;

/**
 * Created by Abboudi_Aliwi on 02/11/2015.
 */
public class Main extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

    }

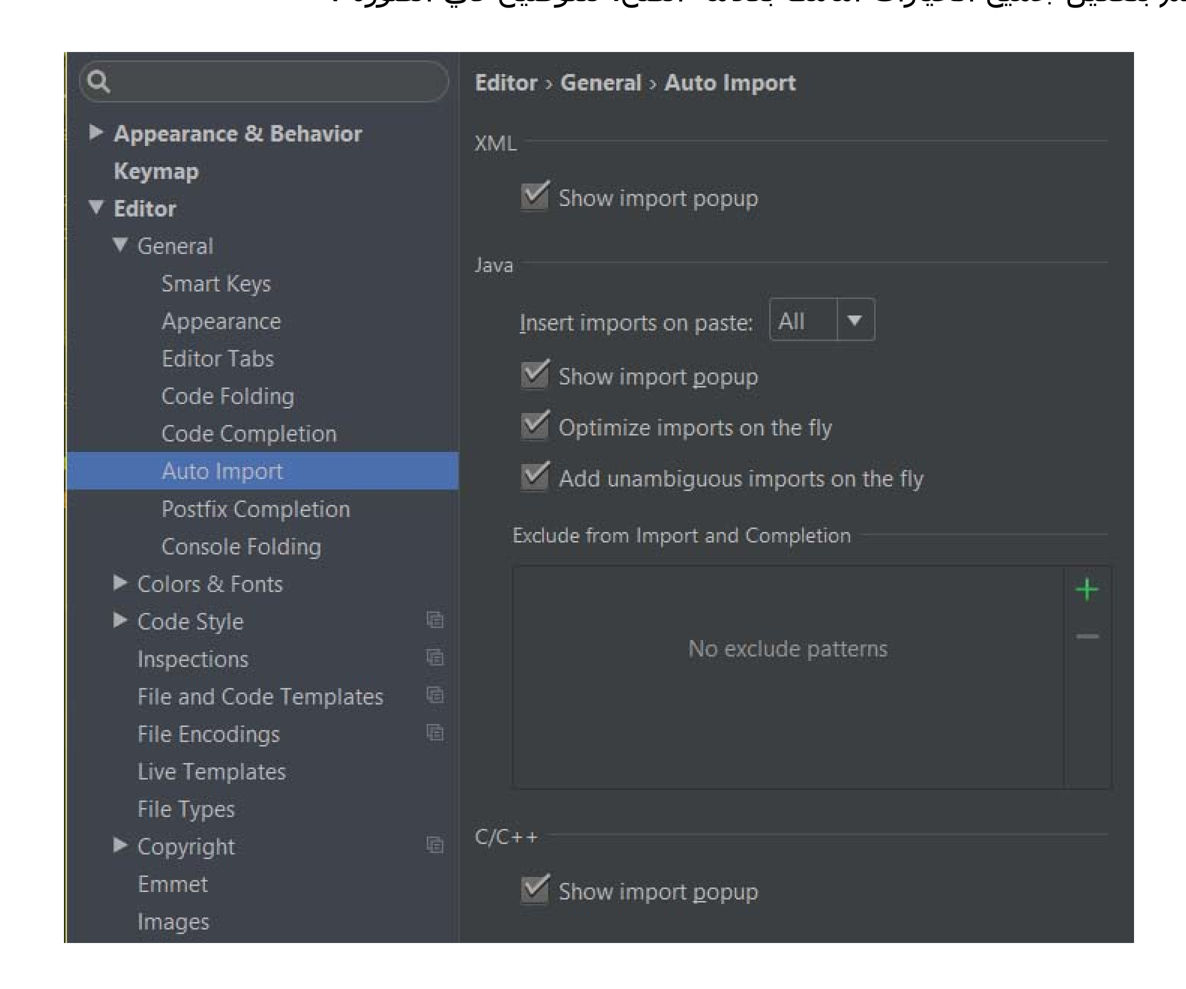
}
```

اما هذه الوضعية فهي مناسبة جداً لمن يريد التركيز على الكود واستخدام البرامج الأخرى بكل حرية, اي سهولة التنقل, لتفعيل هذا الوضع من القائمة اختر View ثم Enter Distraction-Free Mode و للخروج منه كذلك اختر Exit Distraction-Free Mode.

استيراد تلقائي للمكاتب :

إذا كنت من المبرمجين السابقين على برنامج الإكليبس, او على الاندرويد ستوديو وقد قمت بمشاريع ويوماً ما قمت بعملية نسخ أكواد من موقع مثلاً الى البرنامج سوف تجد الكثير من الكلمات لونها أحمر إي أخطاء, ويعني إنها تحتاج إلى مكاتب لدعمها, وتحتاج للضغط على واحد منها وتعمل له استيراد داخل الاكتفتي, إما بهذه الطريقة فسوف يتم الاستيراد تلقائياً إلا في حال بعض العناصر له اكثر من مكتبة لذلك تجعل الاستيراد لك لتختار ما تفضل فيوجد فرق بالتأكد من احداها للآخرة. عموماً لتقوم بميزة التفعيل التلقائي, في الخيارات في الأعلى اختر File ثم Settings. سوف تظهر لك نافذة صغيرة فيها الاعدادات الخاصة بالبرنامج. من الجانب الأيسر اختر Editor ثم General ثم Auto Import .

وقم بتفعيل جميع الخيارات امامك بعلامة الصح. للتوضيح في الصورة :



ترتيب الكود البرمجي :

Main.java

```
package com.andrody.test;

import android.app.Activity;import android.os.Bundle;

/**
 * Created by Abboudi_Aliwi on 02/11/2015.
 */
public class Main extends Activity {@Override protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);setContentView(R.layout.main);

    }

}
```

قبل ترتيب الكود

Main.java

```
package com.andrody.test;

import android.app.Activity;
import android.os.Bundle;

/**
 * Created by Abboudi_Aliwi on 02/11/2015.
 */
public class Main extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

    }

}
```

بعد ترتيب الكود

لاحظ الفرق .. أحياناً تقوم بكتابة الأكواد بسرعة .. ولا تهتم لترتيبها .. لذلك البرنامج يتولى هذا الأمر عنك فقط بالضغط على Ctrl + Alt + L .

++ برنامج الأندرويد ستوديو بحر واسع, دخلت القليل في شروحاته ووجدته جداً عميق ونادراً من يحصى جميع اضافاته ومميزاته لذلك تتعلمها من خلال البرمجة عليه, وسوف نذكر المعلومات الهامة من خلال البدأ ببرمجة التطبيقات.

أمور يجب عليك معرفتها قبل البدأ ببرمجة تطبيقات الأندرويد

سوف نقوم ببرمجة تطبيقات الأندرويد بلغة الجافا Java.

الأكواد البرمجية بلغة الجافا تكتب في ملفات .java ونطلق عليها كلاس, الكلاسات منها أنواع مختلفة, ومنها الاكتفتي الذي يرتبط بشاشة او واجهة يراها المستخدم ويتفاعل معها. إذا سوف نطلق على الملف البرمجي بداخله اكواد الجافا ومرتبط باللياوت اسم – Activity -.

اللياوت Layout هي الواجهة التي تظهر للمستخدم, ومبنية بلغة xml إذا سوف نستخدم xml للتصميم والتنسيق و الجافا java للبرمجة إي اعطاء الأوامر.

كل عنصر أو كائن مثل الزر او الصورة او النص إلخ, له خصائص منها الطول والعرض والحجم إلخ.

كل عنصر نريد استخدامه برمجياً واعطائه أوامر يجب إن نضع داخله خاصية id لنقوم بربطه بملف الجافا من خلال هذا المعرف.

لا تضع معرف id متكرر نفس الاسم لأكثر من عنصر.

كل لياوت او واجهة تريد اظهارها للمستخدم يجب أن يكون لها اكتفتي مرتبط بها.

كل اكتفتي في مشروعك يجب إضافته في ملف اسمه AndroidManifest.xml سوف يتم شرح الملف ولماذا نضع فيه الاكتفتي في درس منفصل.

الإسماء التي تضعها للعناصر او الملفات او الاكتفتي او اللياوت إلخ, تجنب استخدام غير الأحرف الإنجليزية, كذلك لا تبدأ الإسماء بالأرقام, ولا تستخدم الفراغات.

تستطيع إثناء كتابة الأكواد استخدام الإكمال التلقائي ليظهر لك خيارات إكمال للكود الذي تحاول كتابته ولو كان كلمة واحدة , استخدم Ctrl + Space.

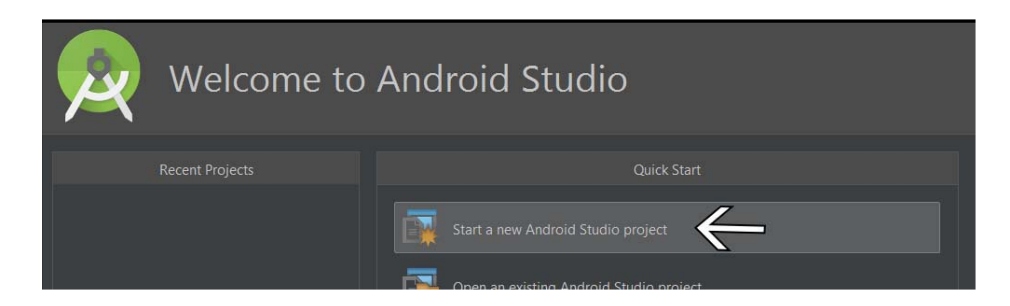
لا تستغرب من قولنا الأب والأبن في العناصر, فلغة xml تتبع النظام الشجري آباء وأبناء, وتستخدم هنا مصطلح Parent للأشارة إلى الأب أو العنصر الأكبر الذي يحتويه العنصر المقصود.

يوجد الكثير من المكاتب الجاهزة التي تضيفها لمشروعك لتستفيد منها في بعض المزايا.

يوجد شركات او مواقع تقدم API أو SDK لتستفيد من خدمات الموقع في تطبيقاتك.

إنشاء مشروعك الأول للاندرويد

في هذا الموضوع سوف نتعلم كيف نقوم بإنشاء مشروع جديد فارغ, للبدأ ببرمجة تطبيقات الأندرويد.



في البداية وبعد تشغيل برنامج الأندرويد ستوديو, سوف تظهر لك النافذة الترحيبية, اختر الخيار الأول وهو Start a new Android Studio project. سوف تظهر لك نافذة أخرى لإعدادات المشروع.

New Project
Android Studio
Configure your new project
Application name: MyApplication 1
Company Domain: Andrody.com 2
Package name: com.andrody.myapplication 3 Done
Project location: C:\Users\DELL\AndroidStudioProjects\New\MyApplication 4 ...
Previous Next Cancel Finish

حسب الصورة السابقة :

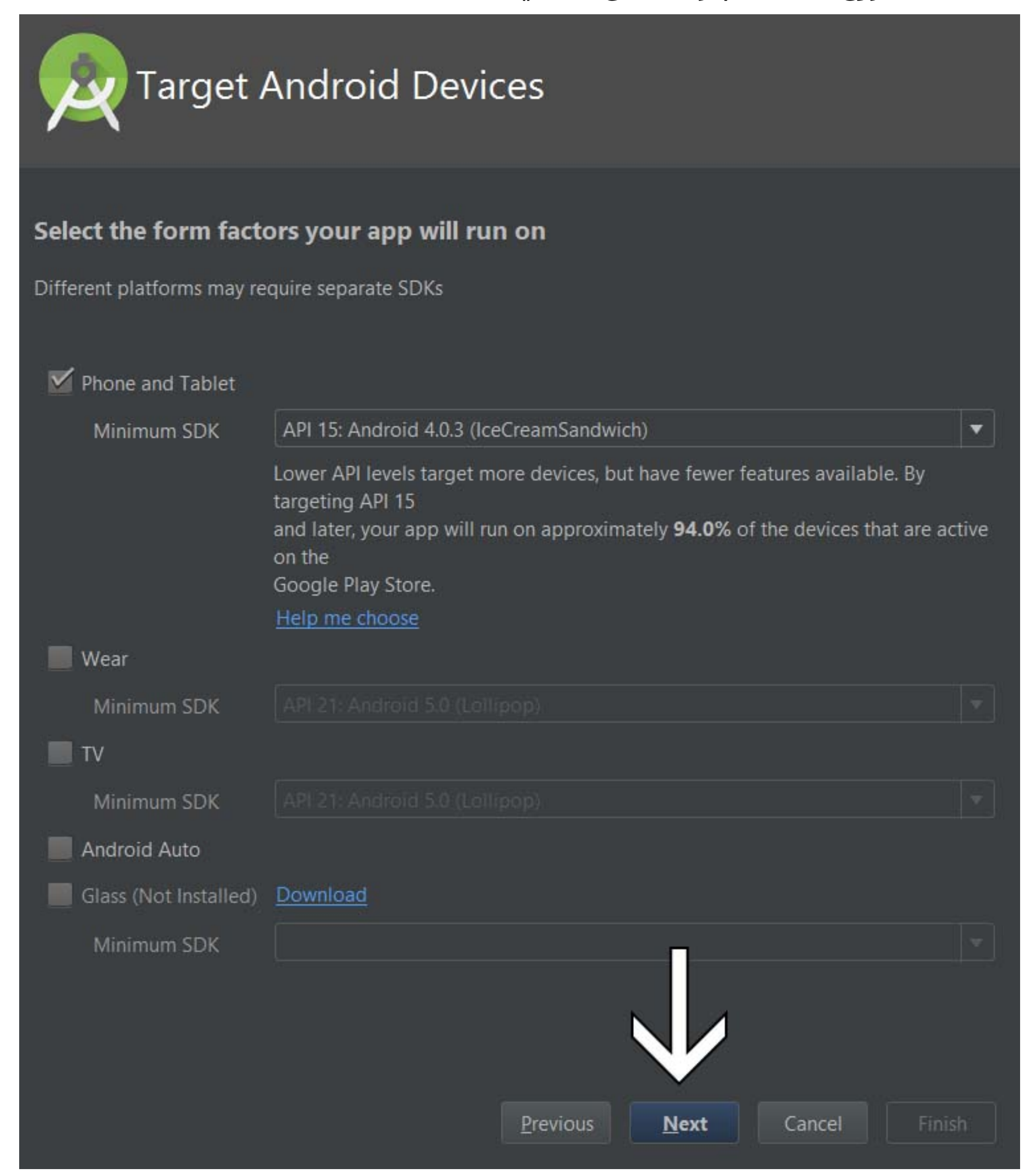
1 / عنوان التطبيق.

2 / اسم خاص بك, والمعروف وضع عنوان موقعك لضمان عدم تشابه بموقع آخر. لأن اسم الموقع واسم التطبيق تقوم بالتأثير بإسم الحزمة – الباكيج -.

3 / معرف التطبيق وهو اسم الحزمة كما ذكرت, هذا المعرف يجب أن لا يتشابه مع إي تطبيق آخر, ومن هذا المعرف نتعرف على عنوان او رابط التطبيق على المتجر. مثلاً تطبيق الفيس بوك على المتجر : https://play.google.com/store/apps/details?id=com.facebook.katana

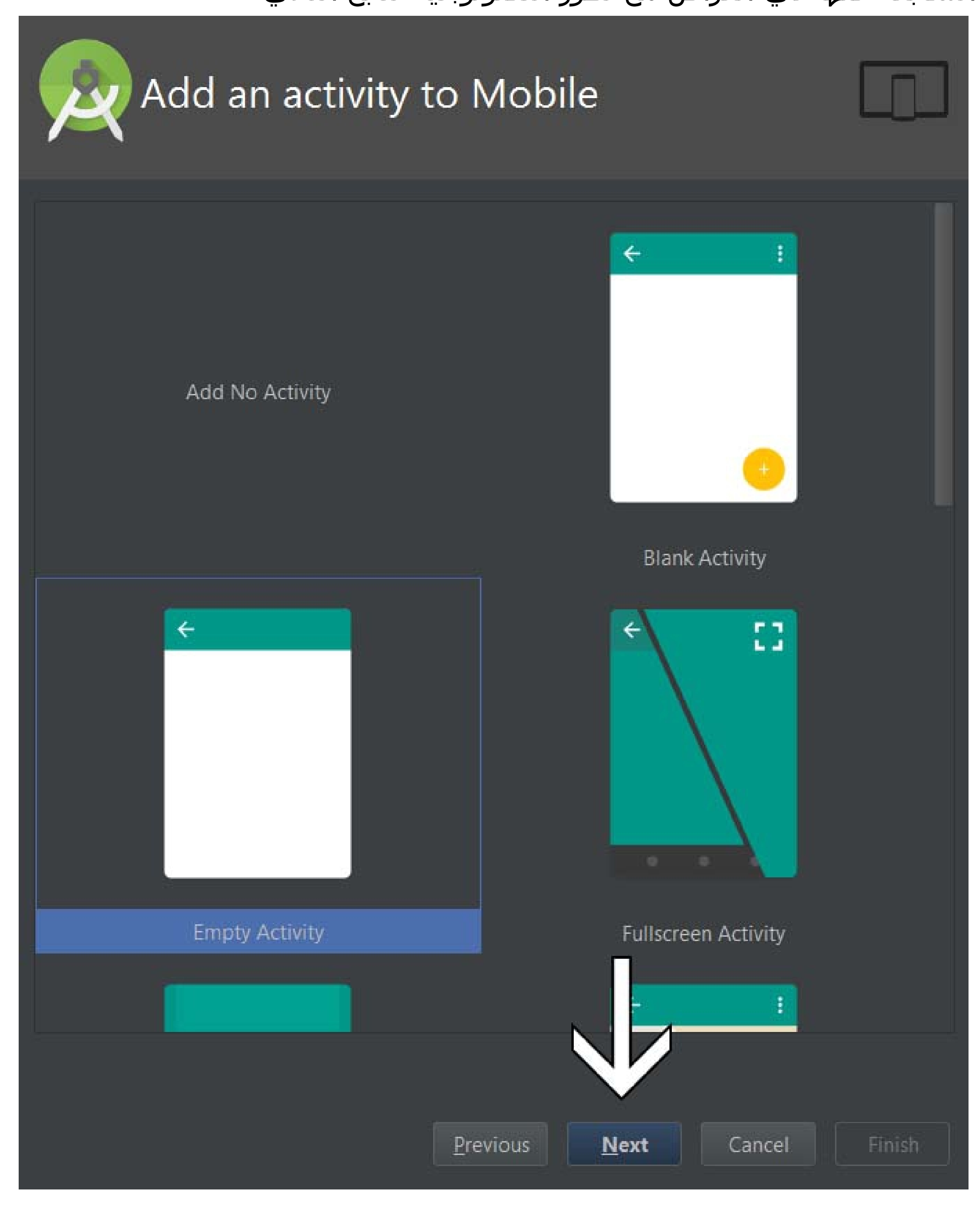
إذا المعرف الخاص بتطبيق الفيس بوك هو : **com.facebook.katana.**

4 / مسار حفظ المشروع على جهازك. نتابع التالي ....

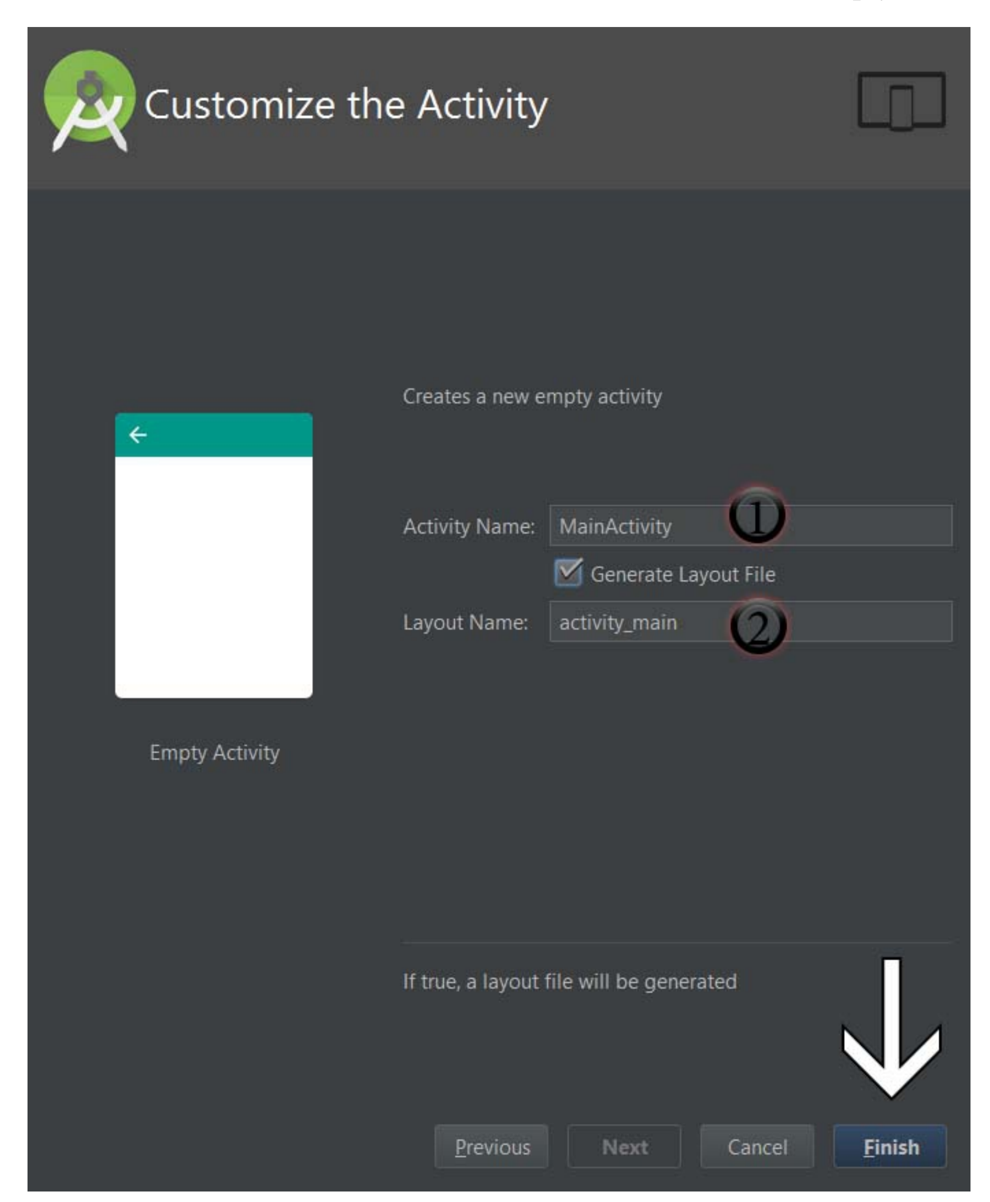


الصورة السابقة, لأختيار نوعية الأجهزة التي يعمل عليه تطبيقك, لأجهزة الجوال والتابلت, والساعة الذكية وشاشة التلفاز و نظام ايتو للسيارات و النظارات الذكية. ما يهمنا فقط لأجهزة الجوال والتابلت وهذا ما سوف يتم شرحه في الكتاب.

أما بما يخص Minimum SDK فهو كما شرحنا سابقاً أقل إصدار للأندرويد يعمل عليه تطبيقك هذا الذي تقوم بإنشائه. وتستطيع الاستعانة بـ الضغط على Help me choose لتظهر نافذة يوجد بها اصدارات نظام الأندرويد وكم عدد الأجهزة لكل نظام وهكذا تستطيع معرفة كم الكمية من الاجهزة في العالم التي تستطيع استخدام تطبيقك, فالصورة موضح API 15 اي نظام 4.0.3 وأعلى فقط يستطيعوا استخدام تطبيقي, والنسبة تقريباً 94% من أجهزة الاندرويد سوف يعمل عليها تطبيقي وهذا ممتاز. فالإصدارات السابقة كلها في انقراض مع تطور التكنولوجيا. نتابع التالي ..



حسب الصورة السابقة فهنا يظهر لنا بعض من النماذج الجاهزة للبدأ بالبرمجة عليها .. ولكن ما سوف نختاره هو Empty Activity. نتابع التالي ...

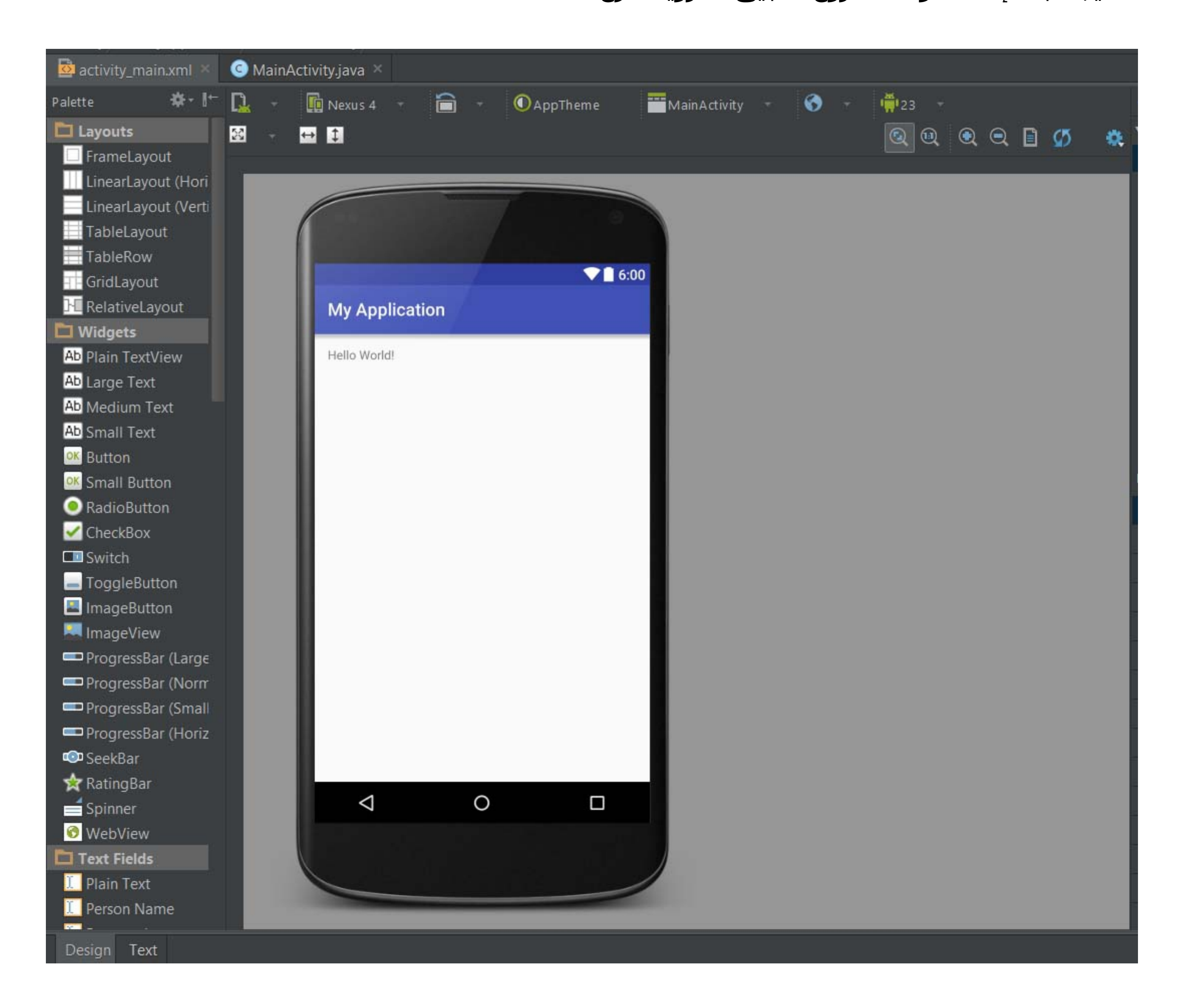


1 / العنوان الخاص بالاكتفتي التي سوف تعمل معك في بداية تشغيل التطبيق. ضع الاسم الذي تريد مع وضع أول حرف كبير – كبتل –.

2 / عنوان اللياوت ومن المتعارف عليه أن تكون نفس اسم الاكتفتي ولكن اول حرف صغير – سمول – وهي اللياوت المرتبطة بهذا الاكتفتي.

حالياً اتركها كما هي و إضغط انتهاء Finish لبدأ البرنامج بتجهيز مشروعك لتبدأ البرمجة عليه.

النتيجة بعد إنشاء أول مشروع تطبيق اندرويد فارغ :



شرح الأكواد الافتراضية التي تظهر لك بداية إنشاء مشروع تطبيق اندرويد

بعد إنشاء أول تطبيق أندرويد لنا, نلاحظ إنه لدينا ملفات جاهزة مفتوحة لنا وهي الاكتفتي التي يتم تشغيلها في بداية تشغيل التطبيق والتي هي كذلك مرتبطة باللياوت أو الواجهة التي سوف تظهر للمستخدم.

سوف نجد MainActivity / الاكتفتي.

ملف كما ذكرنا سابقاً يحتوي على أكواد بلغة الجافا, يتم تنفيذها في بداية تشغيل التطبيق.

والكود يأتي السطر الأول له اسم الباكيج كما حددناه للتطبيق, ومن ثم مكاتب الدعم للعناصر

المستخدمة في هذه الاكتفتي, ومن ثم الكلاس وداخله الدوال.

الكود الافتراضي للاكتقتي الرئيسية :

```
//  اسم الباكيج للتطبيق إي المعرف او الآي دي

 package com.andrody.myapplication;

//  استيراد لمكاتب الدعم وكلما أضفنا عناصر إزداد عددها

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

//  الكلاس واسمه من ثم مشتق من اب كومبات اكتفتي لكي يدعم تصميم الماتريال للاصدارات السابقة وفي الغالب يستخدم مشتق من اكتفتي

public class MainActivity extends AppCompatActivity {

//  دالة الأون كريت وهي الدالة التي يتم تنفيذها في بداية تشغيل هذه الاكتفتي
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
//  اللياوت التي يتم ربطها بهذا الاكتفتي
    setContentView(R.layout.activity_main);
  }
//  هنا نستطيع استخدام دوال أخرى داخل هذا الكلاس والتي لا يتم تنفيذها حتى يتم استدعائها في دالة الاون كريت
}
```

و كذلك **activity_main.xml / اللياوت.**

اللياوت ذكرنا وسوف نذكر مرة اخرى هو ملف التصميم او التنسيق للواجهة, وهذا اللياوت هو الواجهة الرئيسية التي سوف يتم استدعائها من قبل الاكتفتي ملف الجافا الرئيسية اي يعني ان هذه الواجهة هي التي سوف تظهر للمستخدم في بداية تشغيل التطبيق.

الكود الموجود داخل الواجهة اللياوت :

```
  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:paddingBottom="@dimen/activity_vertical_margin"
  android:paddingLeft="@dimen/activity_horizontal_margin"
  android:paddingRight="@dimen/activity_horizontal_margin"
  android:paddingTop="@dimen/activity_vertical_margin"
  tools:context=".MainActivity">
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!" />
</RelativeLayout>
```

لغة XML تستخدم النمط الشجري في التنسيق, إي بمعنى عناصر تحتوي اسفلها او داخلها عناصر

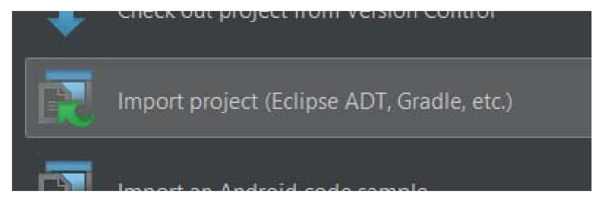
اخرى, بمعنى اب وابناء .. وكما هو ظاهر لنا في الكود لاحظ وسم البداية ووسم النهاية للأب وهي اللياوت الرئيسية من نوع RelativeLayout ويوجد انواع اخرى وهذا النوع اي تستطيع وضع العناصر داخلها بشكل حر في إي مكان على الشاشة. وبداخله عنصر من نوع TextView إي نص ومكتوب داخله عبارة Hello World كما شاهدنا في صورة النتيجة السابقة لانشاء اول مشروع, وأما بالنسبة للخصائص الاخرى لكل العناصر سوف يتم شرحها بشكل مستقل.

أما الآن سوف نتعلم بعض الأساسيات الآخرى, من ثم نقوم بإنشاء مشاريع صغيرة فقط لتوضح لك المبدأ بهذا المجال وأنت تكمل المسير ..

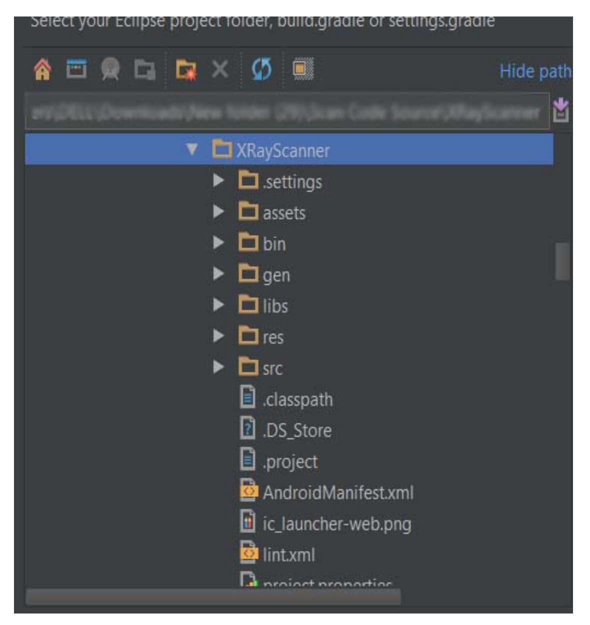
استيراد مشاريع الإكليبس للأندرويد ستوديو

الكثير منا يقوم بتحميل المشاريع الجاهزة, فالمشاريع الجاهزة على الأنترنت أغلبها تم بنائها على برنامج الإكليبس, لذلك من المهم جداً تعرف كيفية استيراد مشاريع الأكليبس مع العلم إنها جداً سهلة ولكن الكثير يتساؤل عن الكيفية ..

في البداية عند تشغيلك لبرنامج الإندرويد ستوديو في النافذة الترحيبية اختر Import project (Eclipse ADT, Gradle, etc.) كما في الصورة :



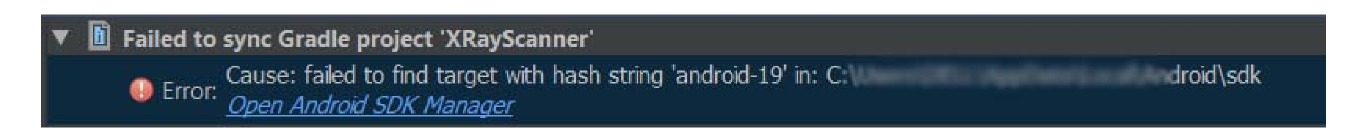
سوف تظهر لك نافذة صغيرة لأختيار مجلد المشروع, تقوم بتحديد المجلد الذي يحتوي على المجلدات src + res وغيرها .



بعد اختيار المشروع, اضغط أوك ومن ثم سوف يظهر لك خيار لتحديد المسار الجديد الذي سوف يضع فيه المشروع بملفاته الجديدة.

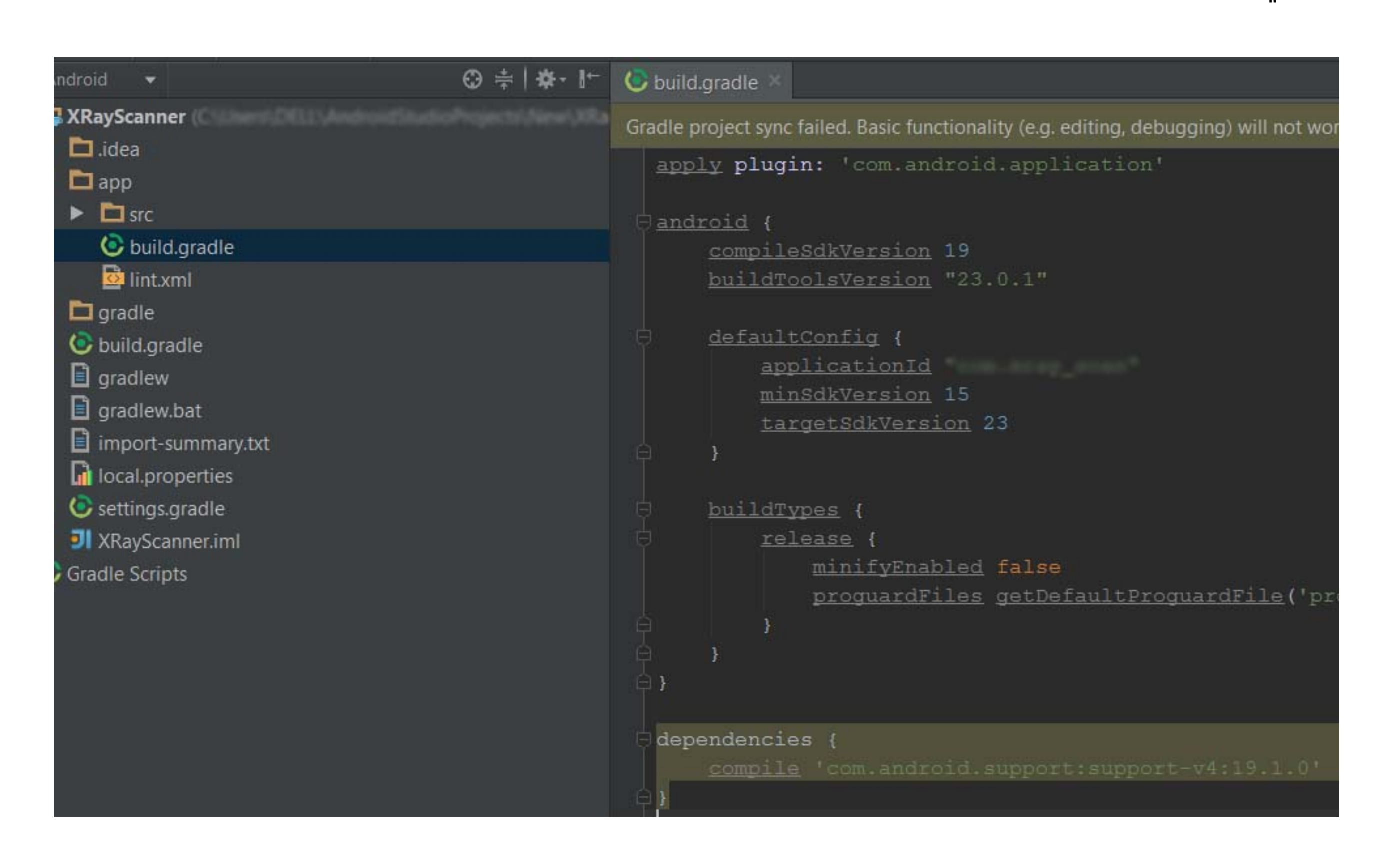
سوف يقوم برنامج الأندرويد ستوديو تلقائياً بتحويل الملفات للشكل الجديد الذي يتناسب معه.

بعد انتهاء المعالجة, في الغالب تظهر لك رسالة خطأ مثل هذه :



وهذا معناه إن المشروع أعلى إصدار فيه هو API 19 وأنت الآن في عصر متقدم أصبحت تستخدم API 23 أو أعلى, لحل المشكلة فقط تذهب للمشروع وتجعله يتوافق مع أعلى أصدار.

تفتح الملف "build.gradle" الذي بداخل مجلد app , وتقوم بتعديل المطلوب, سوف تجده بالشكل التالي :



قم بتعديل 19 إلى الأصدار الذي تستخدمه لديك والأعلى بداخل SDK , وحسب الموجود هو 23.

بعد الانتهاء من التعديل أضغط على زر مزامنة أو في الشريط الأصفر Try Again

ليصبح الملف بهذا الشكل : لاحظ كيف أصبح ترتيب الملفات وألوان النص.

```
Android
▼ app
  ▶ manifests
  ▶ java
  ▶ res
  ▶ resources
  ▶ assets
▼ Gradle Scripts
    build.gradle (Project: XRayScanner)
    build.gradle (Module: app)
    settings.gradle (Project Settings)
    local.properties (SDK Location)
```

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 23
    buildToolsVersion "23.0.1"

    defaultConfig {
        applicationId "
        minSdkVersion 15
        targetSdkVersion 23
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('progu
        }
    }
}

dependencies {
    compile 'com.android.support:support-v4:23.1.0'
}
```

نكون هكذا أنتهينا من عملية الاستيراد, ولكن بعض المشاريع تستخدم مكاتب دعم مختلفة, لذلك ما يجب عليك القيام به, إذا كانت مكاتب جوجل لا مشكلة إضافتها من جديد بإضافة سطر واحد فقط بهذا الملف, ولكن يوجد مكاتب لا يتم استيرادها ويعطي خطأ برنامج الاندرويد ستوديو, لذلك تستطيع فتح المشروع على اكليبس وإلغاء المكاتب بشكل مؤقت, من ثم تعمل استيراد على الأندرويد ستوديو إرجاع المكاتب للمشروع إن كنت تحتاجها.

مجلدات وملفات المشروع

لو لاحظتم بعد إنشاء المشروع الأول على الأندرويد ستوديو ظهرت لنا الملفات والمجلدات التالية لمشروعنا >> كما في الصورة.

```
Android
▼ app
  ▼ manifests
      AndroidManifest.xml
  ▼ java
    ▼ com.andrody.myapplication
        MainActivity
    ▶ com.andrody.myapplication (androidTest)
  ▼ res
      drawable
    ▶ layout
    ▶ mipmap
    ▶ values
▼ Gradle Scripts
    build.gradle (Project: MyApplication)
    build.gradle (Module: app)
    proguard-rules.pro (ProGuard Rules for app)
    gradle.properties (Project Properties)
    settings.gradle (Project Settings)
    local.properties (SDK Location)
```

بالنسبة لـ مجلد Values + ملف AndroidManifest.xml سوف يتم شرح كل منها بشكل مستقل.

مجلد java /

المجلد الذي يحتوي على الباكيج, تستطيع إضافة أكثر من حزمة داخل مشروعك, ولكن سوف يتم تشغيل واحدة منها الرئيسية والآخرى ممكن تحتاج أستخدامها في مشروعك.

الباكيج يحتوي على ملفات الجافا, كلاسات منوعة, تستطيع كتابة كلاسات على راحتك, مثلاً كلاس يحتوي على نصوص فقط لتستخدمها في كامل المشروع, وهكذا.

والمهم لدينا إنه الاكتفتي داخل الباكيج, وهذه الاكتفتي بلغة الجافا هي التي سوف نتصل منها باللياوت او بالواجهات لنبدأ بإعطاء الأوامر البرمجية للتطبيق وللأدوات والعناصر إلخ.

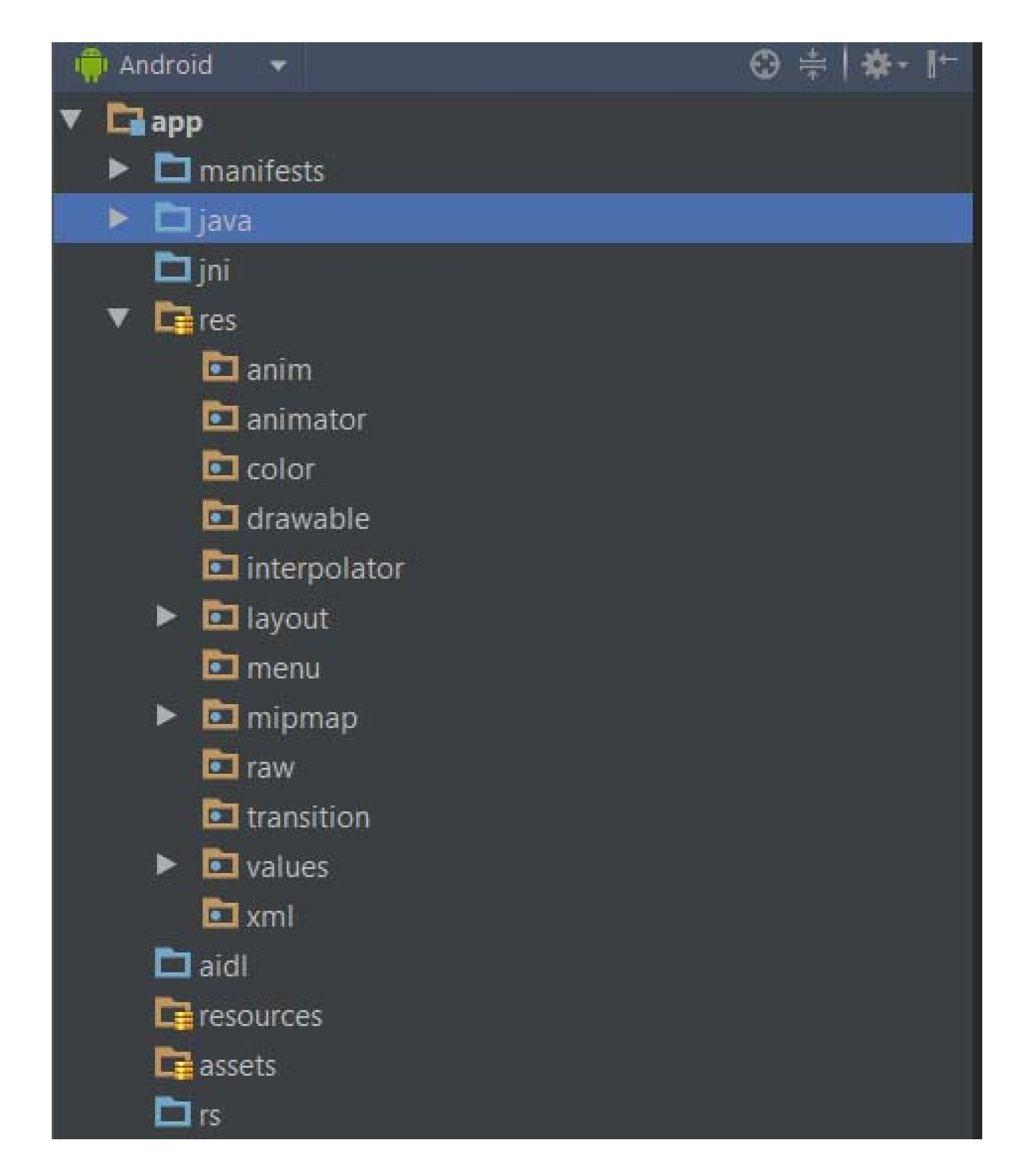
مجلد res /

بمعنى resources الموارد او المصادر التي تستخدمها في تطبيق, بمعنى آخر كل ما يراه المستخدم أمامه أثناء تصفح التطبيق, الواجهات, قائمة الخيارات, ايقونة التطبيق, الصور المستخدمة في التطبيق, الصوتيات, الحركات او التأثيرات, الألوان, النصوص, الثيم أو الاستايل إلخ

وهو يحتوي على عدة مجلدات لتوزيع هذه الموارد, بالشكل التالي :

- **drawable /** المجلد الذي تضع فيه الصور التي تريد استخدامها في التطبيق + ملفات xml للتنسيق, مثلاً تأثير الزر بعد الضغط عليه, أو خلفية للتطبيق متعددة الألوان, أو تكرار صورة باترن للخلفية إلخ.
- **layout /** المجلد الذي يحتوي على ملفات اللياوت أو الواجهات التي تظهر للمستخدم وهو من أكثر المجلدات التي سوف تستخدمها. وسوف نكرر كثيراً في شرحنا له باللياوت.
- **mipmap /** يتم تلقائياً وضع أيقونة التطبيق في هذا المجلد, كما تستطيع استخدامه لوضع الأيقونات التي تستخدمها مثلاً زر قائمة الخيارات زر البحث إلخ.
- **values /** سوف يتم شرحه بشكل مستقل.

اختياري لك, إنشاء مجلدات أخرى بهذا الشكل :



سوف أركز على المهم منها فقط, وبشرح مختصر.

من المجلدات التي تستطيع إنشائها في مجلد **res** مثال :

raw : لوضع داخله ملفات الميديا.

menu : لوضع داخله ملفات قائمة الخيارات.

anim : لوضع داخله ملفات التأثيرات أو الحركات.

color : للتحكم بطبقات الألوان مثلاً لون الزر أحمر إذا تم التركيز عليه يصبح أصفر وبعد الضغط عليه أزرق.

xml : في الغالب داخله ملفات التكوين والإعدادات, مثلاً اعدادات للتطبيق.

أما **مجلد assets /** فهو مهم للكثيرين, فيه تضع المواد الخام, صور, صوت, ملفات, قواعد بيانات, صفحات الويب إلخ. وأكثر ما يتم إضافته أنواع الخطوط التي تريد استخدامها في التطبيق.

وما يعيب مجلد assets إن جميع ما بداخله يستطيع إي شخص الوصول إليها وسرقتها :(

أما بقية المجلدات غير مهمة, تحتاجها في مراحل متقدمة, مثلاً لجعل تطبيقك اون لاين مع لوحة جوجل للاعبين وغيرها.

وبالنسبة لـ ملف التكوين فهو أكثر من مهم : **Gradle**

build.gradle / ما يهمنا Module: app والذي بداخله نضع قيم أساسية للمشروع, وهو قام بنصف المهام التي كانت لملف المينيفيست في الأكليبس. مثلاً داخله المعرف الخاص بالتطبيق + أقل وأعلى إصدار أندرويد يعمل عليه تطبيقك و يجب الإشارة داخله لجميع المكاتب التي تستخدمها لدعم مشروعك.

شرح مجلد الفايلوس Values

مجلد الفايلوس هو مجلد يحتوي على القيم والمتغيرات .. النوع:الاسم:القيمة مثلاً نضع داخله قيمة من نوع لون واسمها أحمر والقيمة هي كود اللون الأحمر #ff0000 وهكذا ..

ما يحتويه مجلد الفايلوس هي ملفات xml لكل منها استخدام معين, مع العلم إنك تستطيع دمجها او جعلها كلها في ملف واحد أو تغيير اسمائها .. فقط ملفات الفايلوس يهم ما بداخلها للوصول إليها, وسوف نتعرف كيف يتم الوصول لها في درس مستقل (الوصول لموارد المشروع).

من أهم الملفات التي يحتويها مجلد الفايلوس :

- strings.xml : ملف القيم أو النصوص, وهو أكثر من مهم حالياً ما تجده داخله اسم التطبيق, تستطيع تغييره من داخله, ومن ثم لك حرية إضافة النصوص التي تريد.

والفائدة منه مثلاً لدينا قيمة داخله بإسم Abboudi والقيمة لها عبدالقادر, إذا تستطيع في المشروع استخدام نص سترنج Abboudi وسوف يتم طباعته على إنه عبدالقادر وهكذا.

والسترنج في الريس res على عدة أنواع ثلاثة : String وهو للنص المفرد, أما لسلسلة من النصوص فنستخدم String Array وأما للكميات والجموع فنستخدم Plurals.

وهذا كود يوضح الفرق بين **كلاً منها** :

```
// String
    <string name="string_name">text_string</string>

// String Array
    <string-array
        name="string_array_name">
        <item>text_string</item>
    </string-array>

// Quantity Strings (Plurals)
    <plurals
        name="plural_name">
        <item quantity=["zero" | "one" | "two" | "few" | "many" | "other"]>text_string</item>
    </plurals>
```

- styles.xml : ملف الاستايل او الثيم المستخدم للتطبيق, في الغالب نضع استايل معرف في ملفات الاندرويد وهو يعطي القيم الخاصة به لمشروعنا, وتستطيع الاستفادة منه في أمور كثيرة منها التحكم بالحجم النص بمشروعك والالوان إلخ. وسوف نتعرف عليه بموضوع مستقل.

- colors.xml : ملف لوضع قيم للألوان واستخدامها في مشروعك, مثلاً نضع مجموعة الوان ونضع لها اسماء, من ثم نستخدم فقط الاسم في المشروع ويتم التعرف على القيمة التي وضعناها.

وغيرها من الملفات التي سوف تتعلم وتتعرف عليها من خلال استخدامها, والأنواع التي يخرجها مجلد الفايلوس هي :

Bool – Color – Dimension – ID – Integer – Integer Array – Typed Array

ويوجد الكثير من الملفات والمجلدات تستطيع الاستغناء عنها, وذلك بإستخدام ملفات اخرى تقوم بنفس المهمة. ولكنها وضعت من أجل الترتيب.

أهم ما يميز مجلد الفايلوس, إنك عندما تضع فيه جميع ما تستخدم في مشروعك, تستطيع تكرار مجلد الفايلوس للغة اخرى, وتغيير القيم داخله وهكذا تستطيع برمجة تطبيق متعدد اللغات, وسوف نشرح كيفية بناء تطبيق متعدد اللغات في موضوع مستقل.

شرح ملف AndroidManifest.xml

الملف هذا كان يمتلك صلاحيات كثيرة, ولكن مع الأندرويد ستوديو أصبح يتقاسمها مع ملف build.gradle و الآن هذا الملف هو جداً مهم تستطيع إعتباره إنه ملف وثيقة المشروع. لا يصلح المشروع من دونه.

وتستطيع استخدام داخله وسوم كثيرة, لكن سوف نذكر أهم ما يوجد داخله, وهي إننا نقوم بوضع إي اكتفتي لدينا بالمشروع, لو لم نضعها كلها, لن يتم التعرف على غير الموجودة, وفي حال تم تطلبها بالمشروع سوف يتم التطبيق عن العمل. لأن لم يتعرف على غير الموجودة داخله. وطريقة إضافة الاكتفتي جداً سهلة, أبسطها :

```
<activity android:name=".Activity" />
```

بنفس الطريقة مع وضع اسماء جميع الاكتفتي لديك, الاكتفتي وليس الكلاسات :)

و كذلك في هذا الملف نضع الصلاحيات التي نريدها من تطبيقنا, مثلاً الوصول للملفات, الوصول والاتصال بالانترنت, استخدام الكاميرا إلخ من الصلاحيات المختلفة.

مثال صلاحية الوصول لحالة الشبكة :

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

طبعاً هذه الصلاحية فقط, تحتاج إلى الأكواد لكي تتحقق من حالة الاتصال, الأكواد لا تعمل من دون الصلاحية, لأنها تعمل ولكن تطبيقك لم يخبر الجهاز مسبقاً قبل التثبيت انه يحتاج للتحقق من حالة الشبكة.

من هذا الملف نستطيع أن نحدد ما هي الاكتفتي التي سوف تعمل في بداية تشغيل التطبيق.

باختصار هذا الملف مهم إن تضع فيه كل ما لديك من اكتفتي وصلاحيات, ومعرف التطبيق كذلك, والسبب مثلاً لو كان تطبيقك حجمه يتجاوز 50 ميغا, فقرائته بالمتجر سوف تصبح جداً صعبة, لذلك يتم قراءة هذا الملف فقط لنتعرف من داخله ما هي ايقونة التطبيق وما اسمه وما المعرف الخاص به وما يوجد لديك من اكتفتي وما هي الصلاحيات التي يحتاجها.. وهكذا بسهولة يتم التحقق من تطبيقك والموافقة على نشره بالمتجر.

الوصول لموارد المشروع

بمعنى كيف تصل إلى صورة مثلاً لاستخدامها في تطبيقك وهي موجودة بداخل مجلد الصور.

من المعروف عند الوصول إلى موقع ما نستخدم عنوان محدد مثلاً www.andrody.com أو لملف معين في الجهاز له مسار محدد.

كذلك تحتاج لمعرفة كيفية الوصول للملفات بداخل مشروعك لكي تستخدمها برمجياً.

بالتأكيد ذكرنا سابقاً إن الموارد جميعها بمجلد اسمه res وبرمجياً نشير إليه بحرف R .

ولكن وضع هذا الشرح المستقل لنعرف بعض النقاط, إن المجلدات يتم الوصول إليها بإسمها عدا مجلد الفايلوس فالذي بداخله تستطيع تغيير اسماء ملفاته أو الإضافة إليها, لذلك يتم الوصول إلى ما بداخله عن طريق ما بداخل الملفات التي بداخله.

كذلك يتم الوصول إلى جميع الملفات بداخل مجلدات المشروع بإسمها دون اللاحقة او الصيغة عدا بعض طرق الوصول المختلفة و مجلد assets يجب أن تحدد صيغة الملف لأنك بداخله تستطيع إضافة ما تريد من ملفات ومجلدات.

مثال :

- لدينا صورة إسمها image.jpg وضعناها في مجلد drawable .

يتم الوصول إليها هكذا : R.drawable.image

- لدينا معرف id لإي عنصر مثلاً اسمه test.

يتم الوصول إليه : R.id.test

- الوصول إلى ملفات الواجهات اللياوت لربطها بالاكتفتي : R.layout.activity_main

وأما بشأن ما بداخل **مجلد Values** :

مثلاً قيمة string اسمها app_name بداخل ملف اسمه test.xml بداخل مجلد values.

الوصول لهذه القيمة : R.string.app_name

لاحظ لم يهمنا لا اسم الملف ولا المجلد. فقط النوع والأسم.

وأما بشأن الموارد بداخل **مجلد assets** :

لاحظ إن هذا المجلد ليس بداخل res لذلك الوصول إليه يختلف حسب الطريقة التي تستخدمها,

ويوجد طرق كثيراً جداً. تختلف الطريقة بحسب الموقع الذي تحتاج الوصول منه.

مثال : الوصول لصفحة ويب html داخل هذا المجلد, ونحتاج استخدام صفحة الويب هذه بداخل عنصر ويب فيو. من الجافا نصل إليها كسترنج :

"file:///android_asset/name.html"

أو الوصول إلى ملف خط مثلاً إسمه test.ttf إذا سوف نحتاج الخط في تعريف خط برمجياً من الجافا بهذا الشكل :

getAssets(),"test.ttf"

وهكذا على حسب الاستخدام والنوع.

كل ما سبق في هذا الموضوع هو لاستخدام الموارد برمجياً في الجافا, إما لاستخدامها في ملفات xml بالطريقة التالية :

الوصول للسترنج : string/app_name@

الوصول لاستايل : style/AppTheme@

الوصول لصورة : drawable/image@

وهكذا الأمر جداً جداً بسيط .. فقط مع الاستخدام سوف تتعرف عليه بشكل أكبر.

تحرير الاستايل العام للتطبيق

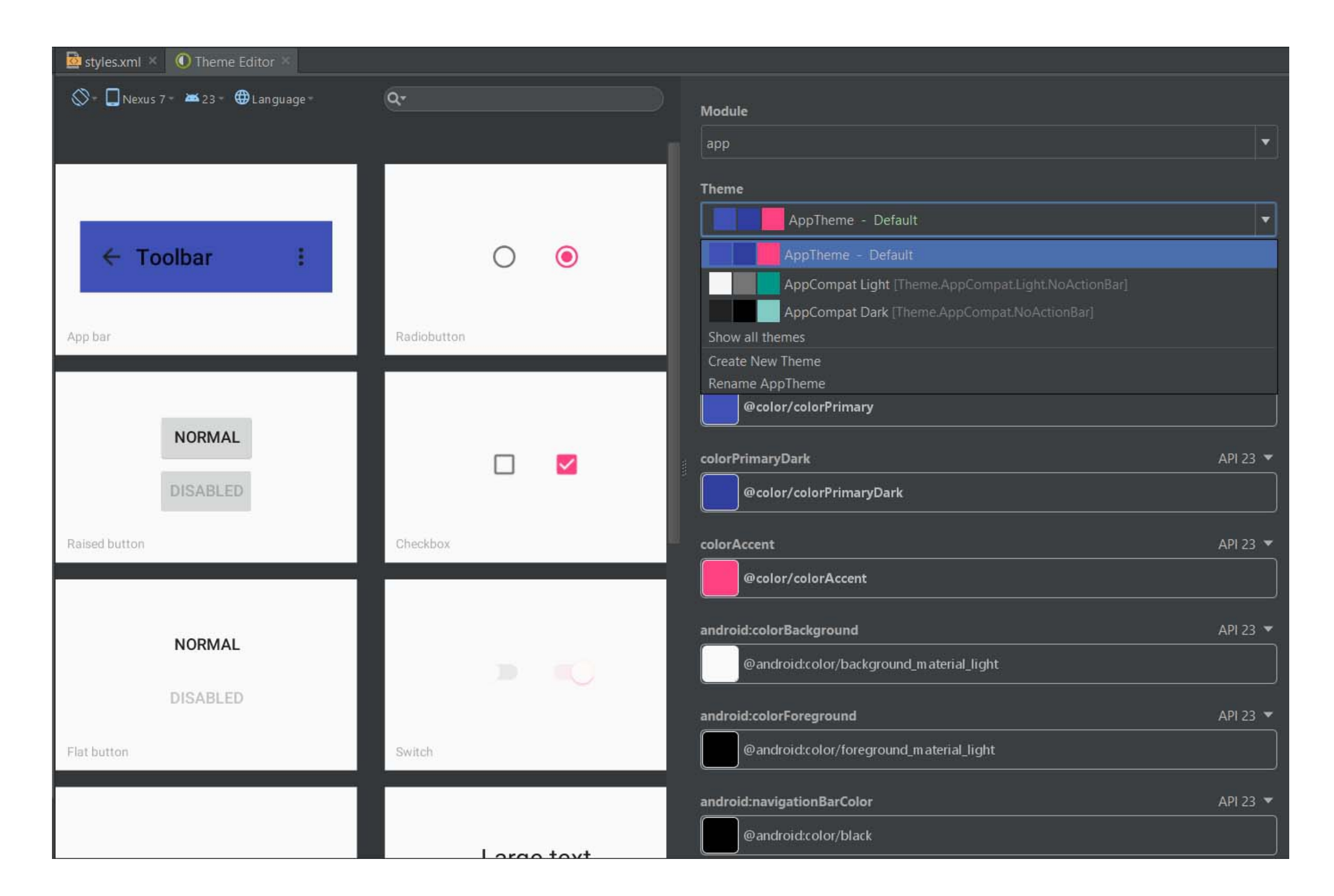
لو لاحظنا ملف الاستايل بعد إنشاء أول مشروع كان بالشكل التالي :

```
styles.xml
Edit all themes in the project in the theme editor.        Open editor   Hide notification
<resources>

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>

</resources>
```

لاحظ في الزاوية العلوية للإخطار يوجد – Open editor سوف يفتح لك المحرر :



تستطيع الاختيار بما يناسبك, وهذا هو تصميم الماتريال ديزاين . وتستطيع معاينة لاكثر من لغة او اصدار والاتجاه إلخ حسب المحمل لديك بالـ SDK من إصدارات وقياسات مختلفة.

ومن لا يعمل لديك الاستايل @style/Theme.AppCompat.NoActionBar

يتأكد من الـ build.gradle إن لديه مضاف مكتبة الدعم :

compile 'com.android.support:appcompat-v7:23.1.0'

حسب الاصدار لديك يتغير الرقم 23.1.0 .

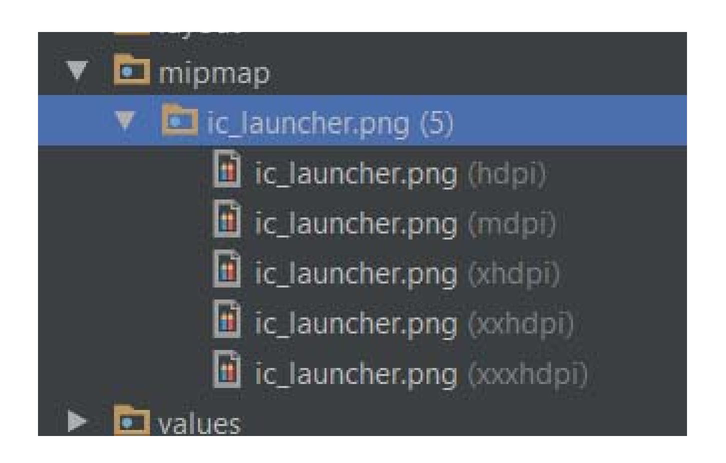
كما يمكنك تعديل الإستايل يدوياً وإضافة أساسيات لكي تستطيع بالتعديل الكامل للتطبيق. وكذلك تستطيع من خلال الاستايل تعديله بالكامل إن كان لجهاز معين أو للغة معينة إلخ ولكن اعتبرها مراحل متقدمة نوعاً ما. إن شاء الله يتم شرحها في كتب لاحقة

إضافة أيقونة للتطبيق

سوف نتعلم الآن كيفية إضافة أيقونة لتطبيقنا الأول كما إنشائه سابقاً, طبعاً تستطيع وضع الصورة بشكل يدوي لكن الأمر متعب قليلاً لتجعل الايقونة متوافقة من حيث الدقة لجميع الاجهزة, فبرنامج الأندرويد ستوديو يستلم عنك ذلك.

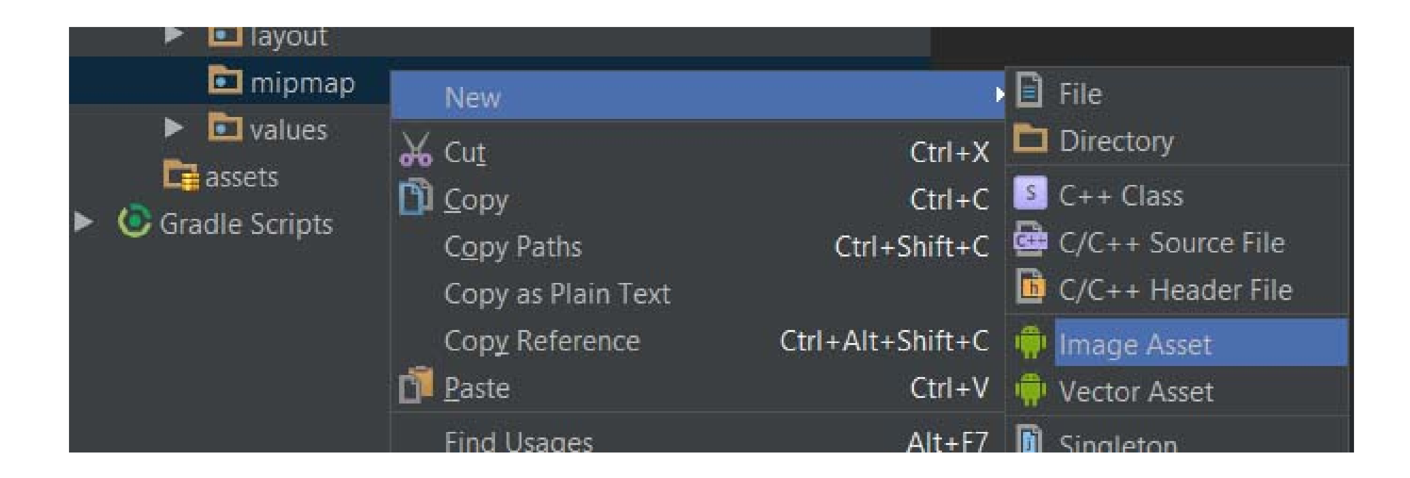
في البداية عند فتحنا لمجلد mipmap كما ذكرنا سابقاً انه المجلد الذي يحتوي على الأيقونات. سوف نجد الأيقونة الافتراضية للتطبيق متكررة خمس مرات كما في الصورة :

هذه الأيقونة تم تكرراها بمختلف الأحجام والقياسات لتتناسب مع مختلف الأجهزة سواء كانت كبيرة أو صغيرة.

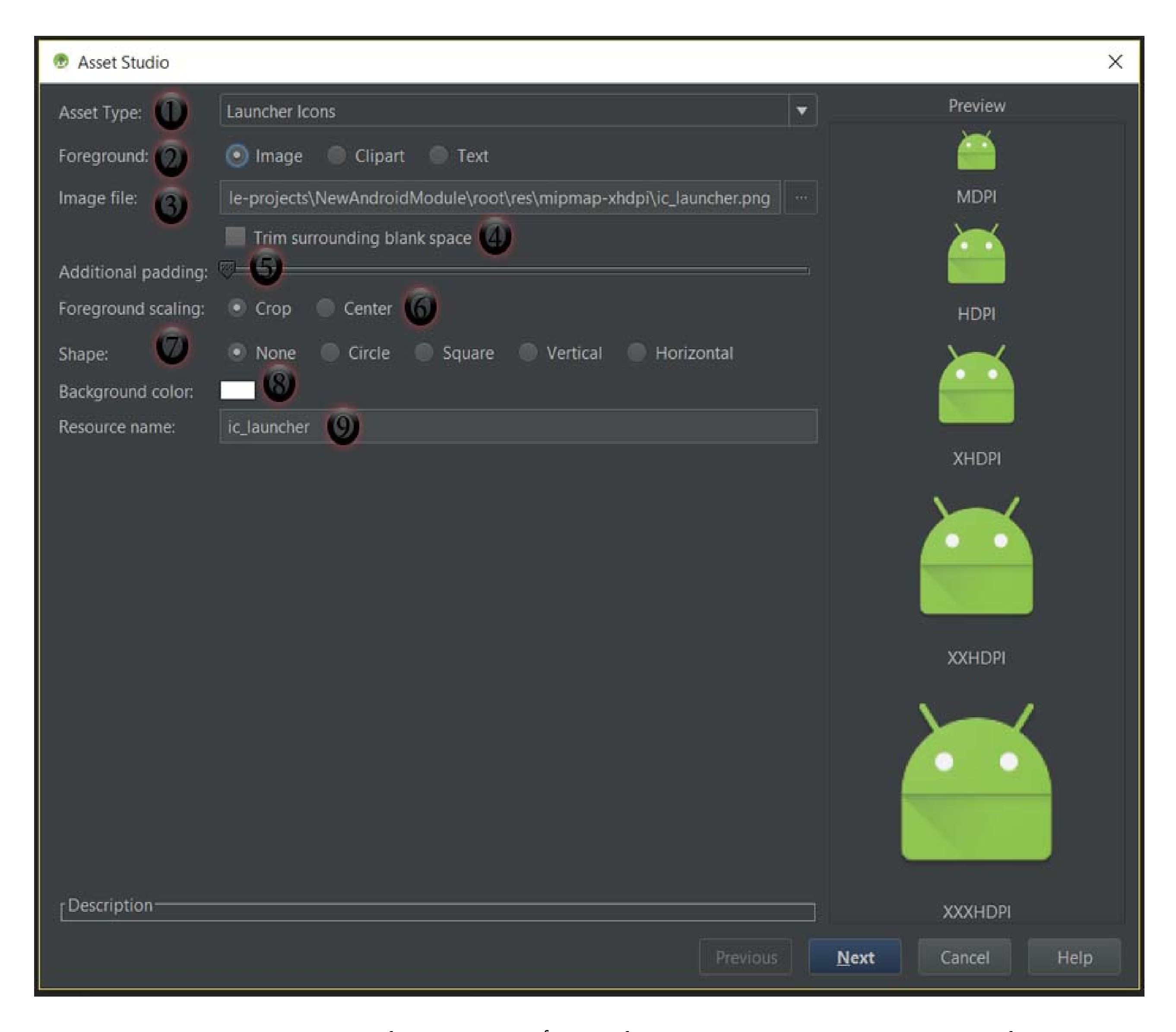


الآن نقوم بحذف هذه الصورة ليتم حذف الخمسة نسخ منها.

بعد الحذف : نقوم بالضغط على المجلد بالزر الأيمن ونختار New ثم Image Asset .



سوف تظهر لك النافذة التالية :



في الجزء الأيمن من النافذة يظهر لك معاينة للأيقونة, أما بالنسبة للأرقام الموضحة :

1 / لماذا تريد هذه الأيقونة ؟ .. هل هي ايقونة للتطبيق, أو ازرار للأكشن بار و التابات أو لشريط الإشعارات.

2 / هل تريدها صورة من جهازك, أو قصاصة فنية تختارها من البرنامج, أو نص ؟! ..

3 / المسار لتختار منه مسار الصورة من جهازك.

4 / إذا كان في مساحة فارغة بالصورة يتم تقليصها.

5 / التحكم بحشو الصورة بمعنى آخر تكبير وتصغير الايقونة.

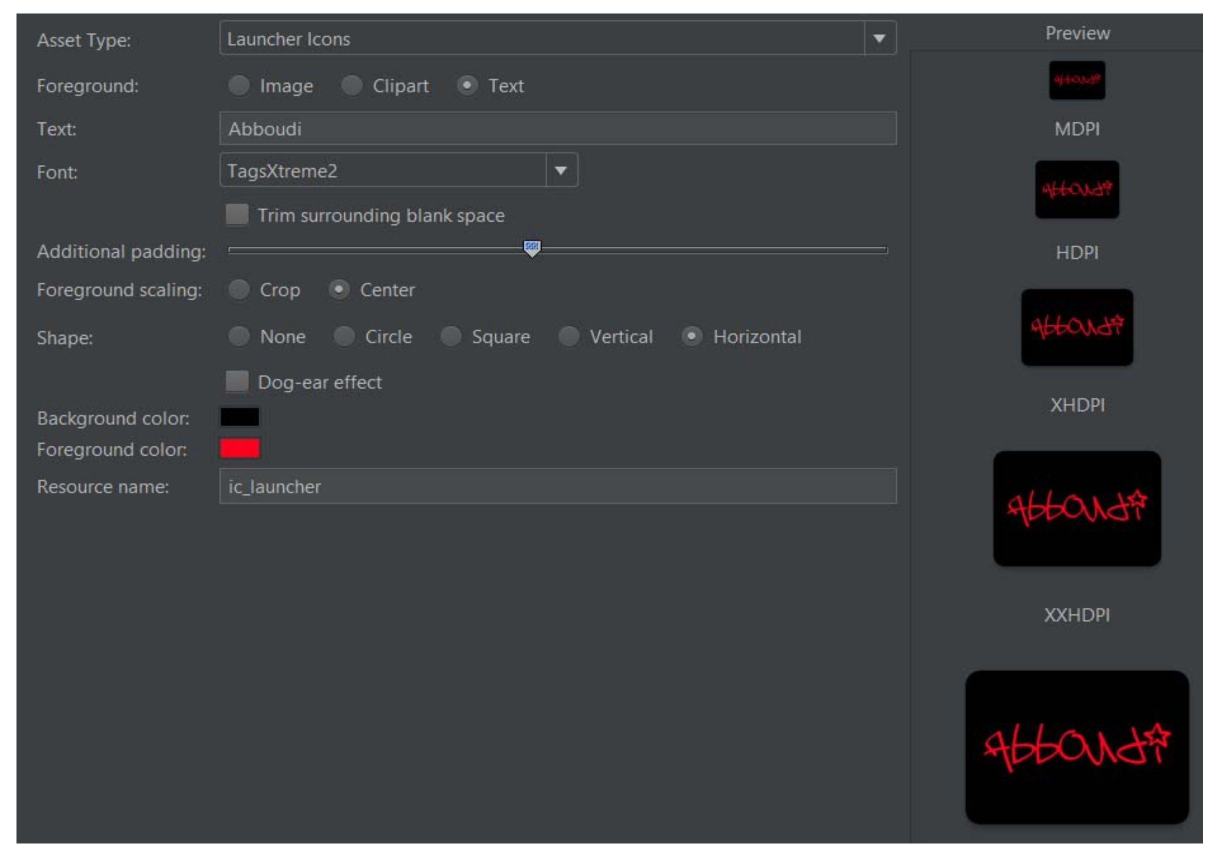
6 / بمعنى لو وضعنا الصورة داخل شكل هل يتم تقليص الصورة لتصبح وسط الشكل او يتم قص الصورة لتناسب الشكل.

7 / لتحديد شكل, دائرة أو مربع إلخ كإطار وداخله القصاصة او الصورة او النص.

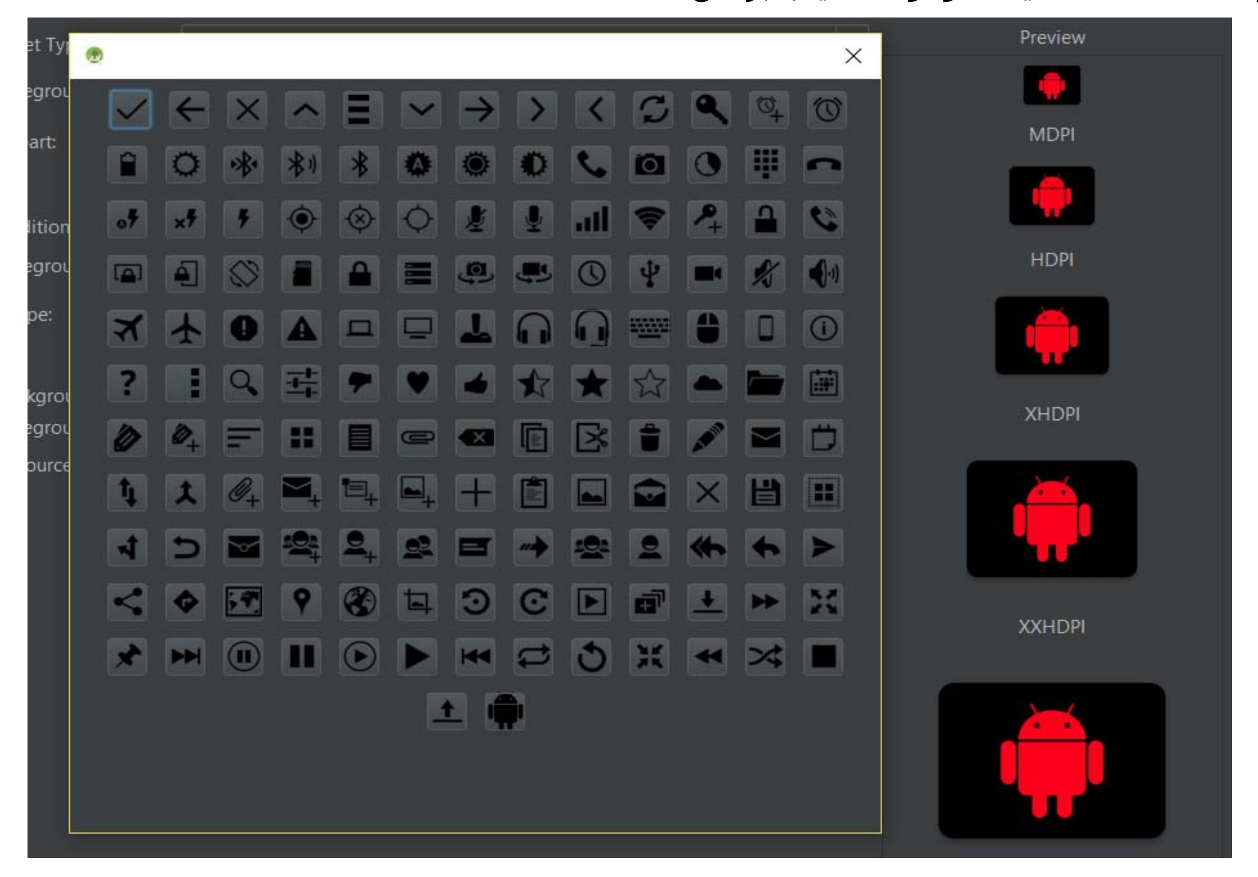
8 / خلفية أو لون الشكل.

9 / اسم الايقونة واتركه كما هو إن أردت استخدم الايقونة للتطبيق.

مع العلم إن جميع الخيارات أحياناً تختلف مع اختيار خيارات اخرى. مثلاً لو استخدمنا النص كمثال :



وأما القصاصات الفنية الموجودة حالياً بالبرنامج :



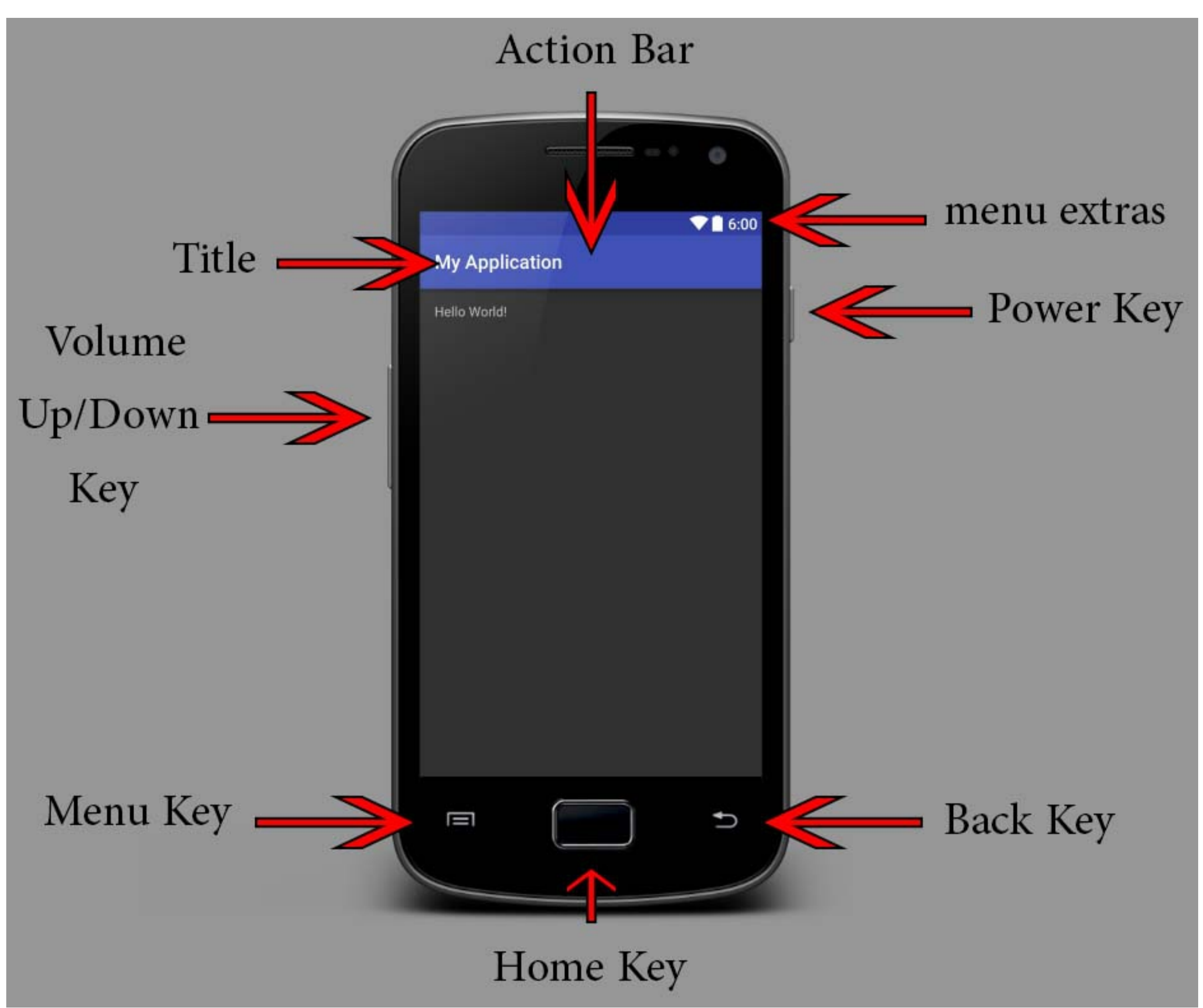
بعد إنشاء الأيقونة التي تريد إضغط على التالي Next ثم انتهاء Finish. وتم الاضافة بنجاح.

التعرف على واجهة جهاز الأندرويد

ما يهمنا معرفته إن في جوالات الأندرويد يوجد فيها ثلاث أزرار خارج الشاشة, إي هاردوير , الأول زر الباك او زر الرجوع للخلف, والوسط زر المنزل او الهوم او الانتقال لرئيسية النظام, والزر الثالث هو زر القائمة, والزر الثالث حالياً في الجوالات الحديثة يبدأ الاستغناء عن مهتمه هذه لصبح زر لعرض نوافذ التطبيقات الأخيرة. ويصبح زر القائمة موجود أساساً في شريط على التطبيق يسمى الأكشن بار.

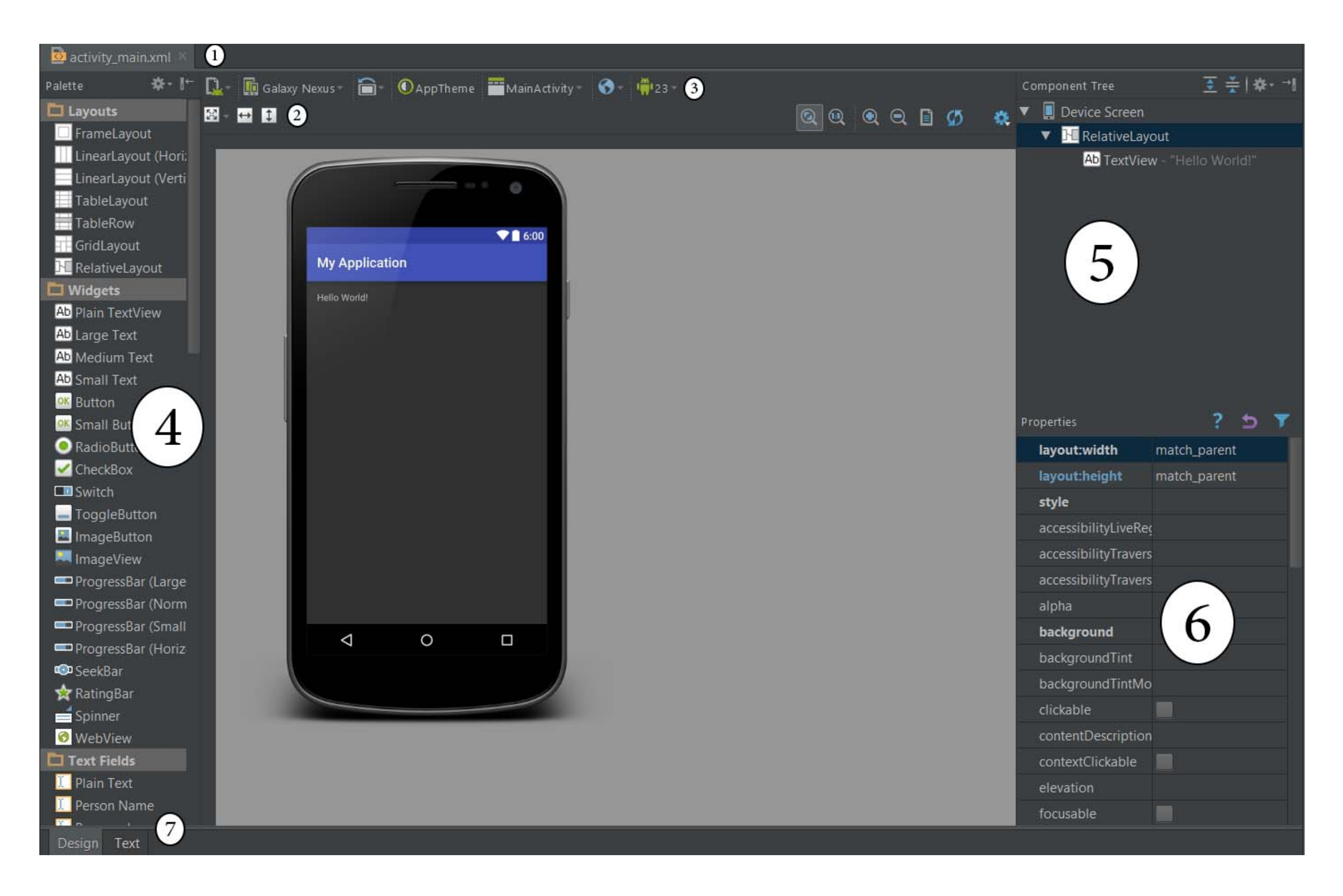
الذي أتكلم عنه جوالات السامسونج ولكن يوجد أجهزة أخرى لا يوجد لديها أزرار خارجية, الأزرار نفسها تصبح على الشاشة. فزر القائمة مثلاً بعض الأجهزة تجده على شريط الأكشن بار والبعض منها في زر خارجي للجهاز والبعض زر على الشاشة في الأسفل أساسي.

ونذكر هذه المعلومات لأننا سوف نتعامل معها برمجياً. فنستطيع إعطاء الأوامر التي نريدها لزر الرجوع ما الذي يقوم به ممكن مهام أخرى, وكذلك زر القائمة وحتى الزر الأساسي للرئيسية.



التعرف على واجهة Layouts

اللياوت أو الواجهة ذكرنا سابقاً إنها تعتمد على لغة XML في البناء ولكن الآن سوف نتعرف كيف نتعامل معها وكيف نقوم بتحريرها وإضافة العناصر . من ثم في الدروس القادمة سوف نتعلم كيف نقوم بمعاينة هذه الواجهات بشكل حقيقي.



1/ عند النقطة الاولى توجد الملفات المفتوحة حالياً.

2/ عند النقطة الثانية يوجد جهة العنصر يمين يسار إلخ, + تكبير/تقليص مساحة عرض وارتفاع العنصر. , ومن الطرف الأيمن للنقطة الثانية, تكبير وتصغير شاشة العرض, وفتح شاشة التكست فيو.

3/ يوجد عدة أزرار, إنشاء جهة أفقية لهذه اللياوت, إنشاء قياسات مختلفة لهذه اللياوت, معاينة هذه اللياوت على مختلف القياسات, معاينة اللياوت على اتجاه مختلف من اليمين الى اليسار والعكس, كذلك معاينتها على مختلف إصدارات الأندرويد, اختيار جهاز معين لمعاينة مؤقتة لهذه اللياوت, تغيير اتجاه اللياوت للاتجاه الأفقي, معاينة العمل على عدة استايلات, ربط هذه اللياوت بملف جافا, ربط وهمي فقط لتتذكر إين هو ملف الجافا المسؤول عن هذه اللياوت, معاينة اللياوت على عدة لغات ومعاينتها على إصدارات إندرويد حسب الإصدارات التي تم تنزيلها لديك في حزم SDK.

4/ العناصر التي سوف تستخدمها في اللياوت, مثل النص والصورة والزر وأنواع اللياوت المختلفة, إلخ.

5/ هنا تجد النمط الشجري للعناصر, وهذا ما قلنا عن أب وأبن ,, تستطيع إضافة العناصر متداخلها بعضها, وأنواع اللياوت من العناصر فقط هي من تسمح إن تضع داخلها عناصر اخرى.

6/ الخصائص للعنصر المحدد عليه, مثل الطول والعرض والنص واللون والحجم والاتجاه والمعرف إلخ.

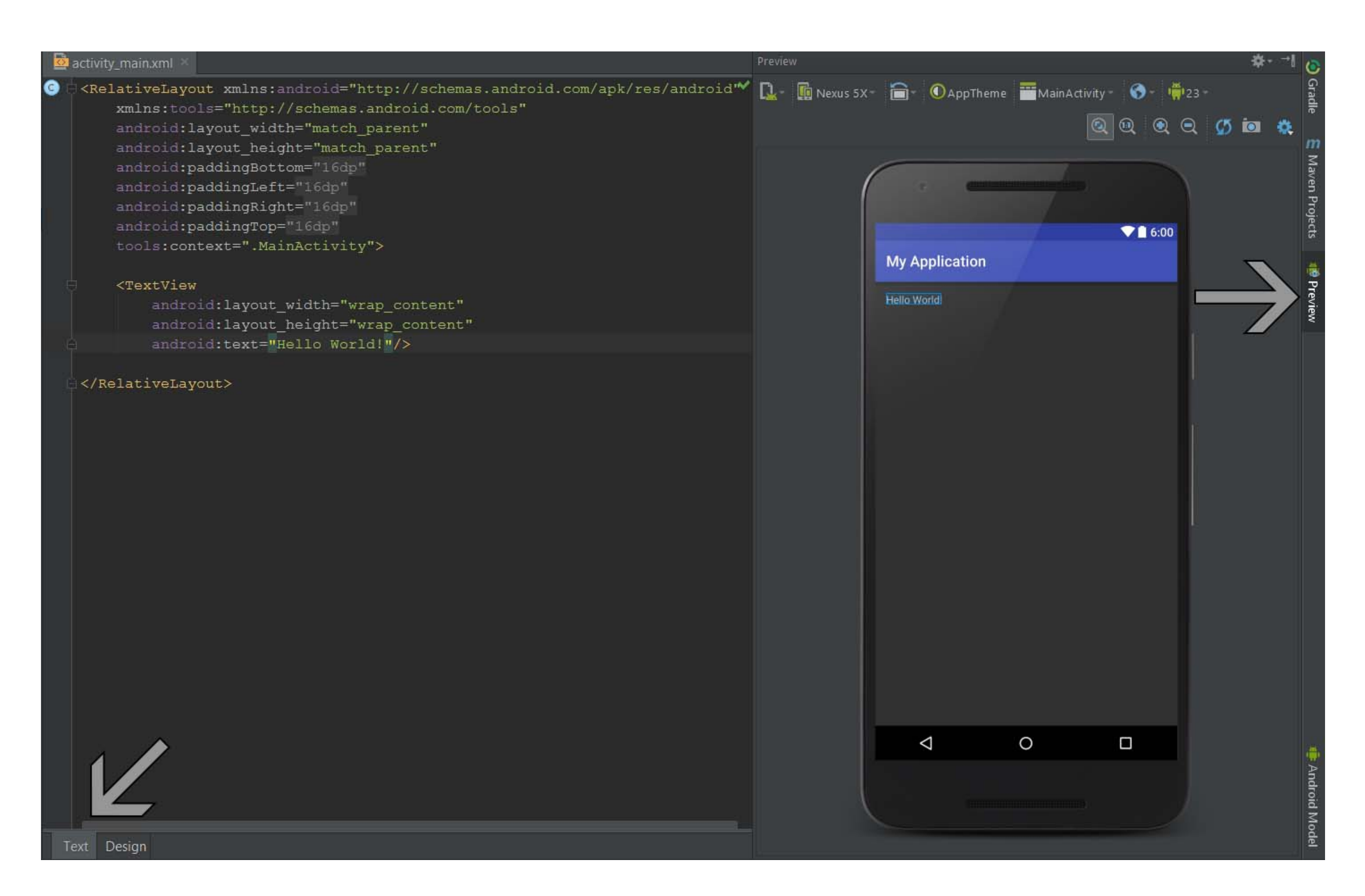
7/ التنقل ما بين شاشة الديزاين فيو والتكست فيو, إي بمعنى شاشة التصميم أو واجهة التصميم والتي هي سهلة للمبتدئين بإضافة وحذف العناصر, إما شاشة التكست فيو فتعتمد على الكتابة أكواد XML. والاثنين لنفس اللياوت, إي تستطيع التعديل من هنا أو هنا كيف شئت.

سوف يتم شرح أغلب النقاط السابقة بشكل منفصل عند استخدامها

عدة نماذج من خلال التحكم بالأزرار الخاصة بالعرض :



بالنسبة لي أكثر وضعية أفضلها, هي شاشة التكست فيو مع المعاينة, تعطيني صلاحية أكبر في التعديل, ومفيدة لمن يعرف الخصائص وكيف يتعامل معها, كما في الصورة :



سوف نتعرف في دروس قادمة على هذه العناصر, وما فائدتها, وما هي الخصائص داخلها, ويوجد خصائص أساسية لبعض العناصر, سوف نتعرف عليها جميعاً .. ولكن أخي القارئ لا تتسرع اقرأ على مهلك وطبق بإحكام, حتى ولو كانت هذه الأمور سهلة وبسيطة, ولكن على غيرك هي جديدة تماماً.

لإضافة العناصر من شاشة الديزاين فيو, فقط قم بسحب العنصر ووضعه على شاشة اللياوت, ومن شاشة التكست فيو قم بفتح <وسم العنصر الذي تريد مع إغلاقه بعد كتابة الخصائص التي تريدها داخله.

إنشاء محاكي AVD

سوف نتعلم الآن كيف نقوم بإنشاء محاكي افتراضي لنستطيع تشغيل عليه تطبيقنا للمعاينة والتأكد من خلوه من إي أخطاء, ومهم جداً لمن ليس لديه جهاز أندرويد. قبل أن تقوم بإنشاء المحاكي من برنامج الأندرويد ستوديو أحب إن أوضح إنه يوجد الكثيرون اشتكو من بطئه وإن يأخذ كمية كبيرة من الذاكرة إلخ, لذلك إذا كان جهازك ليس بالمواصفات العالية, قم بتخطي هذا الدرس وأنتقل للدرس القادم في تثبيت محاكي الجيني موشن واستخدامه في الأندرويد ستوديو فهو يغنيك عن المحاكي الافتراضي من الأندرويد ستوديو.

في البداية نقوم بفتح حزمة SDK ونذهب للأصدار الحالي الذي تعمل عليه أعلى API لديك, وتقوم بتحميل Google APIs ARM EABI v7a System Image هذا أفضلها لك من أجل عدم تحميل دعم لمعالج انتل بالاضافة إلى تشغيل خدمات جوجل بلاي في المحاكي, تحتاجها في تجربة بعض من خدمات جوجل بلاي مثل خرائط جوجل او سحابة جوجل للتراسل إلخ.



بعد الانتهاء من التحميل .. توجه إلى AVD من خلال الأيقونة الخاصة به في الأندرويد ستوديو.

أو من خلال التوجه إلى : < Tools

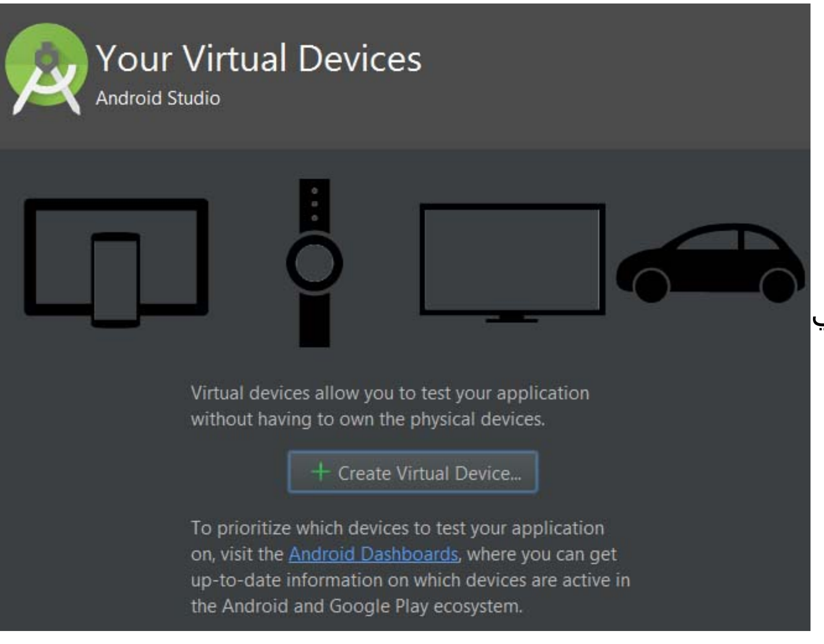
Android < AVD Manager



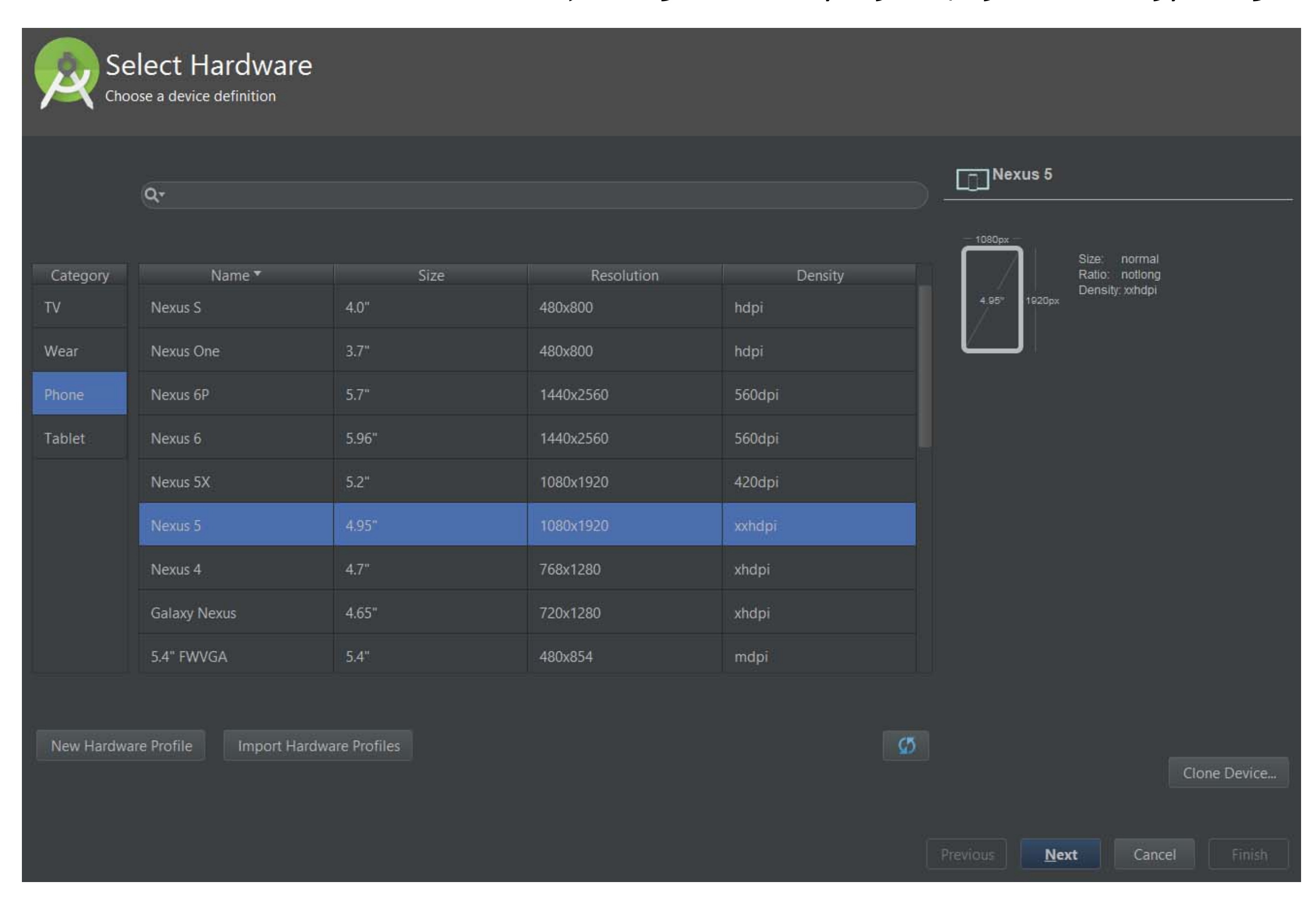
سوف تظهر لك نافذة جديدة كما في الصورة >>

اضغط على على :

Create Virtual Device

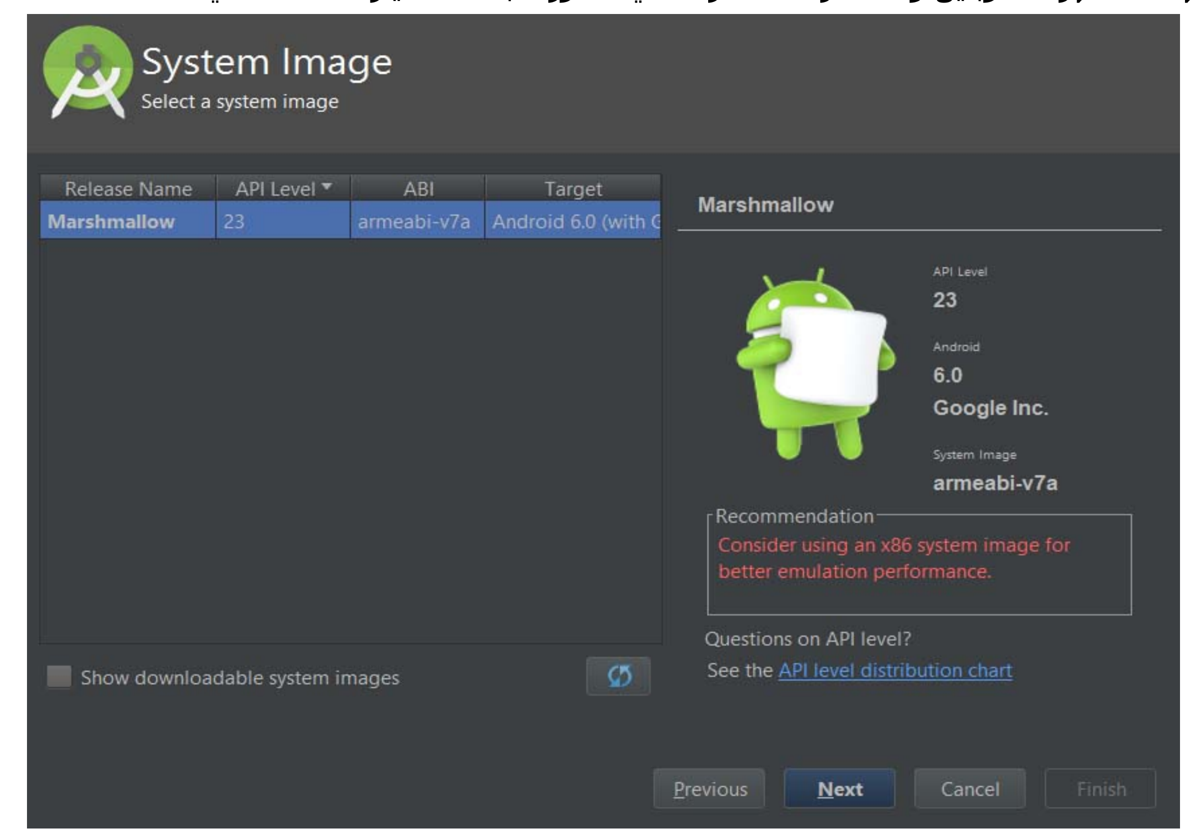


سوف تظهر لك نافذة أخرى, اختار منها الشاشة أو الحجم الذي يناسبك.

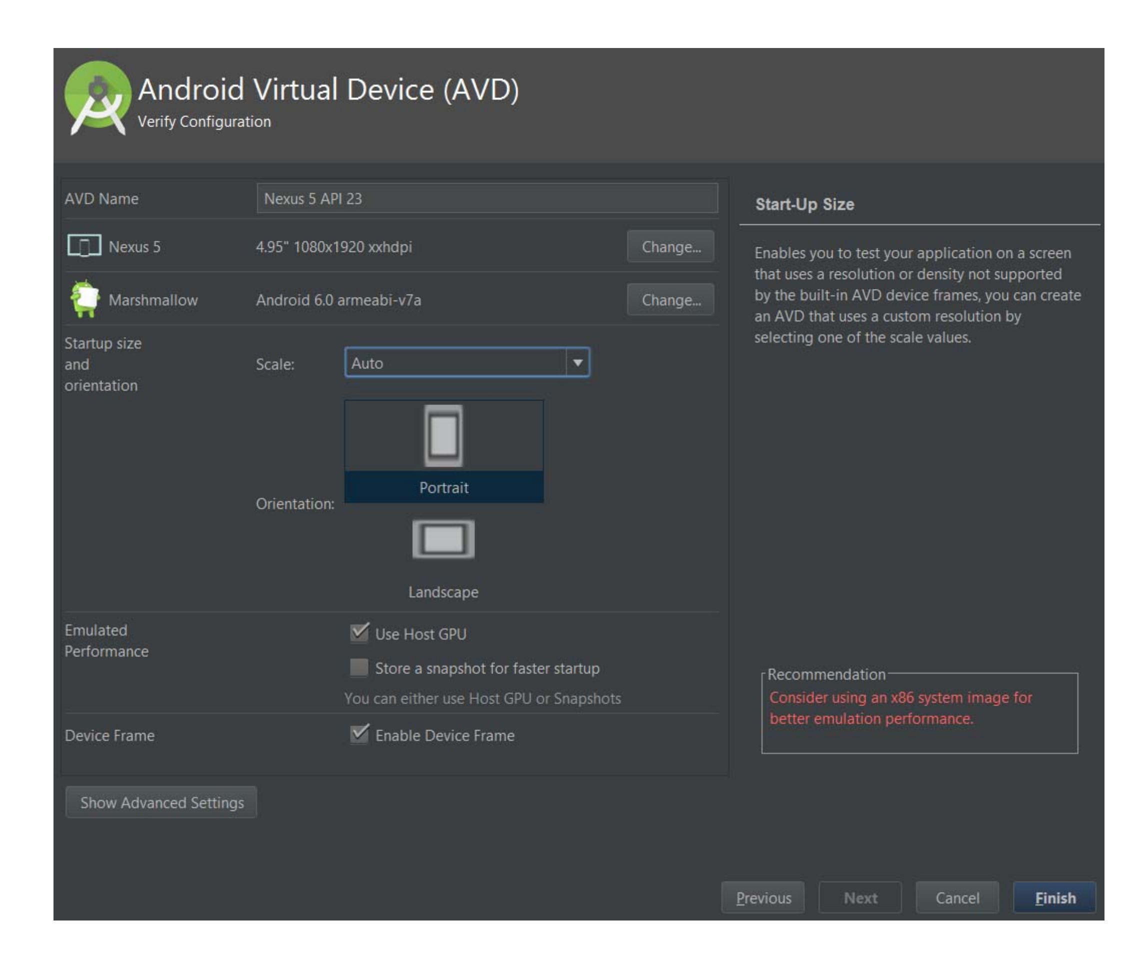


طبعاً لاحظ تستطيع اختيار محاكي لمختلف الأجهزة لمعاينة تطبيقك.

ما يهمنا لأجهزة الموبايل وأنا اخترت كما ترى في الصورة. بعد الاختيار اضغط التالي Next.

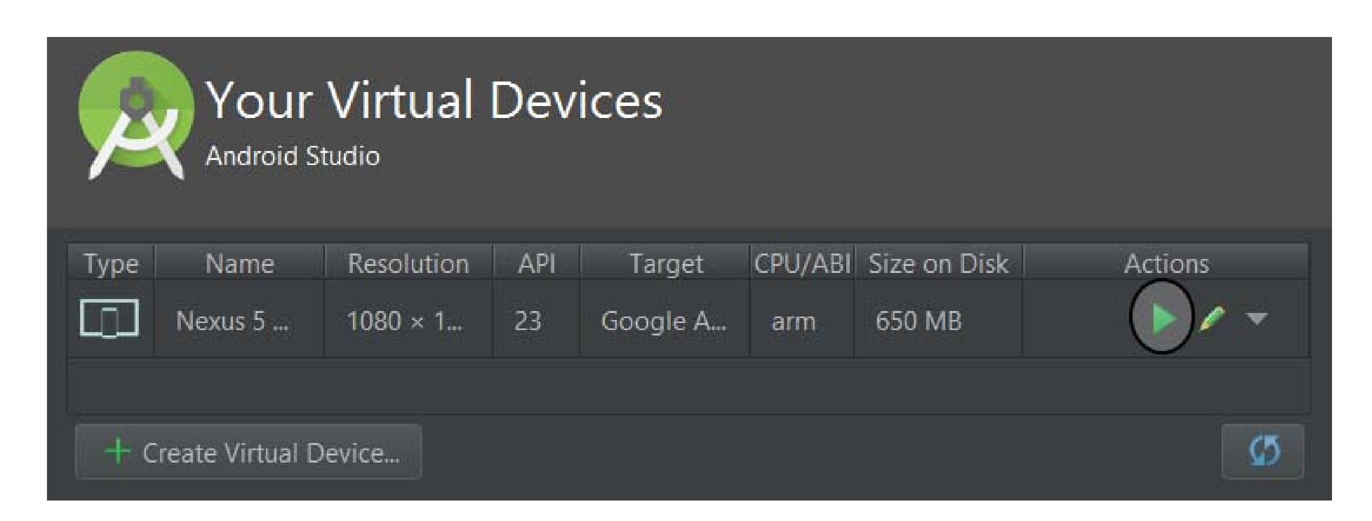


اختر ما قمت بتحميله, واضغط التالي Next.



اترك الخيارات كما هي, واضغط على انتهاء Finish.

سوف تظهر لك النافذة وفيها جميع الأجهزة التي تم إنشائها, تستطيع إنشاء أكثر من محاكي لتجربة تطبيقك على مختلف الأجهزة وأحجام الشاشات المختلفة.



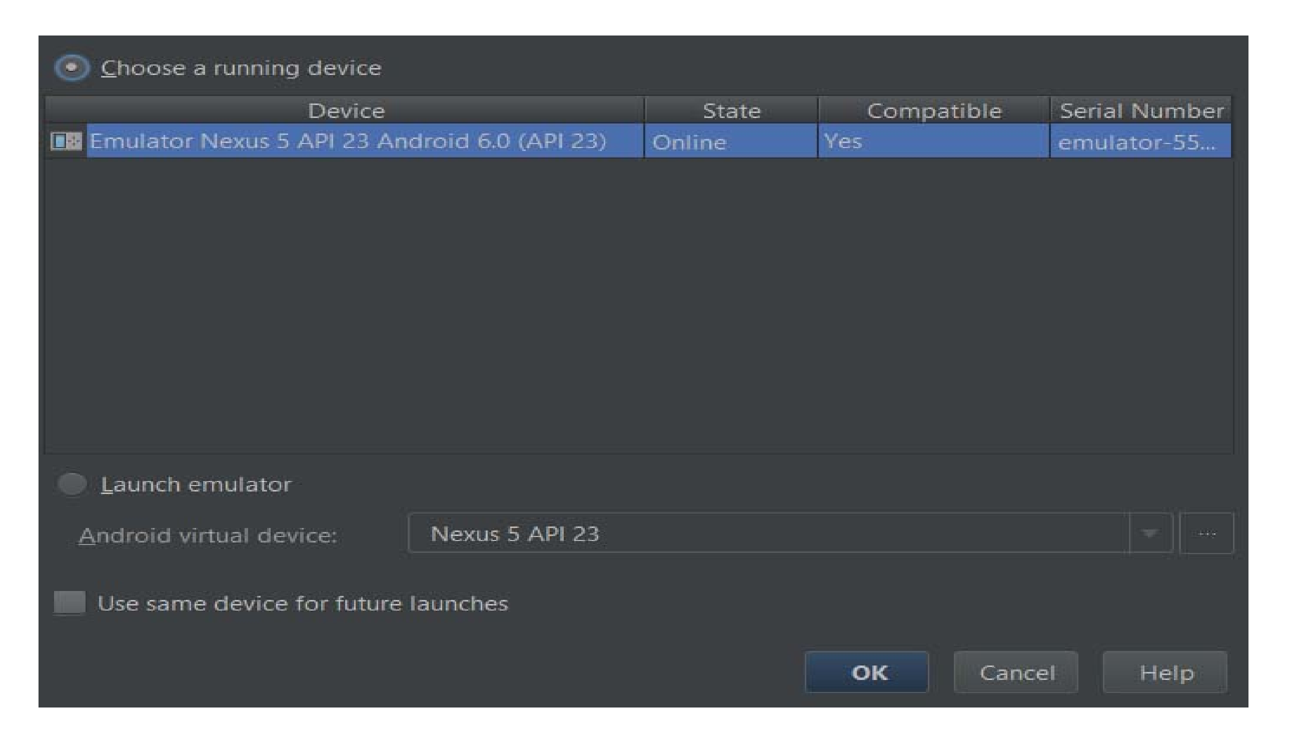
اضغط على المثلث الأخضر, للبدء بتشغيل المحاكي. في أول مرة تشغيل سوف تنتظر كثيراً , وفي كل مرة سوف تنتظر كذلك لكن أقل من المرة الاولى. وذلك حسب سرعة جهازك.

والنتيجة هي بعد إنشاء أول محاكي لنا AVD :

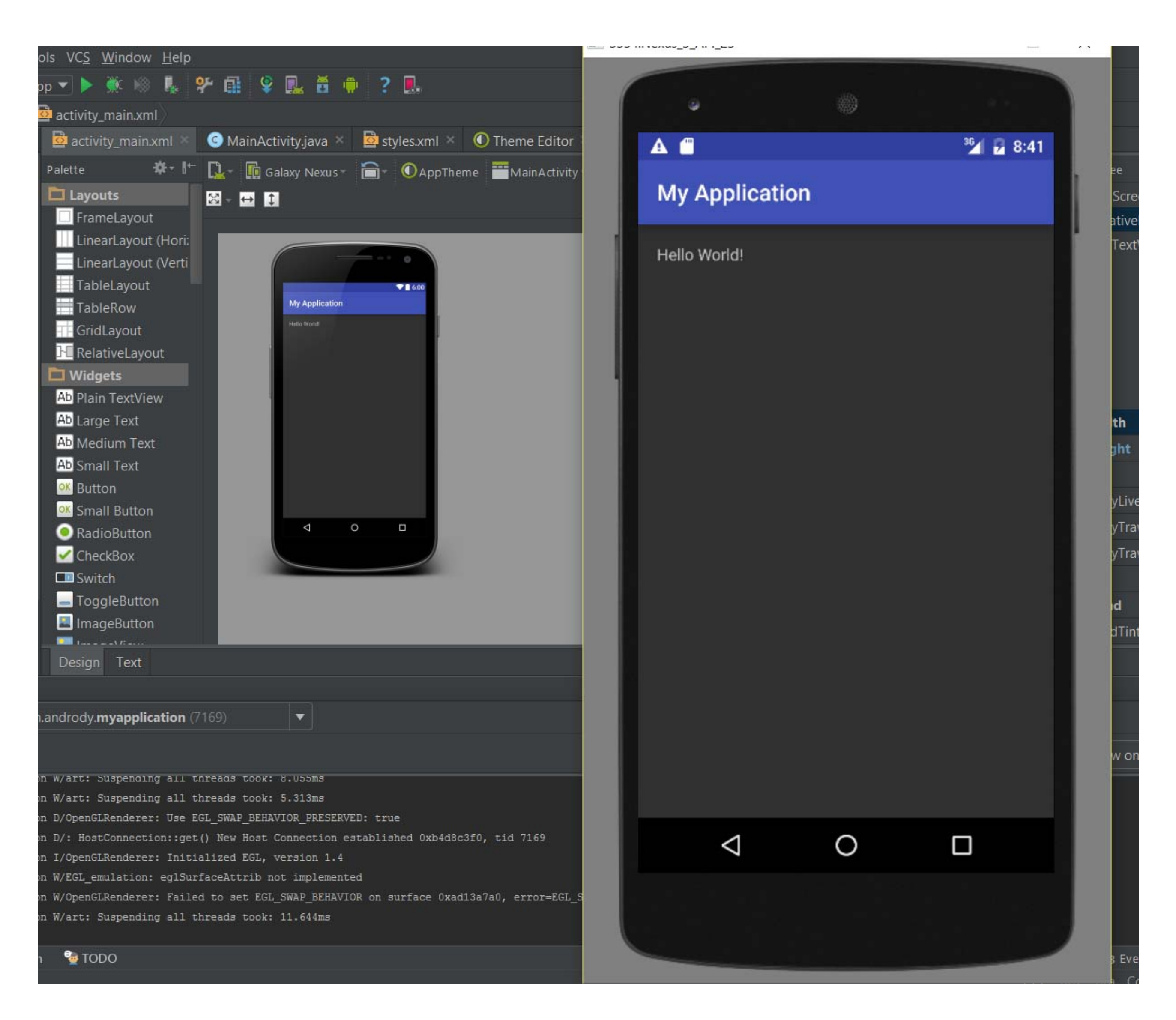


و الآن لتشغيل المشروع الأول كما إنشئناه في الدروس الماضية, فقط نضغط على علامة التشغيل

من شريط الأدوات, مثلث أخضر اللون. بعد الضغط عليه تظهر لك النافذة التالية :



في هذه النافذة جميع الأجهزة التي تعمل الآن وتستطيع معاينة تطبيقك عليها, سواء كانت أميوليتر أو محاكي الجيني موشن أو جهازك الحقيقي usb . بعد اختيار المحاكي الافتراضي نرى النتيجة :



وهكذا نكون قد تعلمنا كيفية إنشاء محاكي افتراضي من داخل الاندرويد ستوديو, وكذلك كيفية تشغيل المشروع عليه.

محاكي الجيني موشن

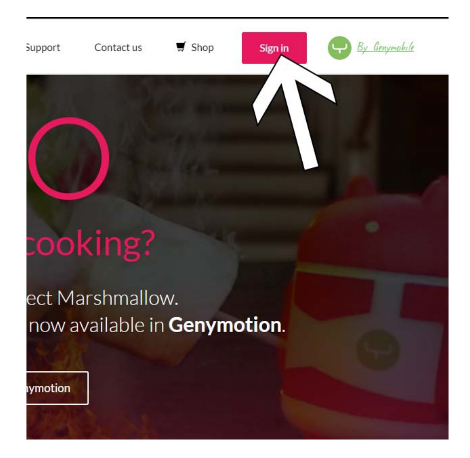
اميوليتر الجيني موشن كما وصف نفسه بإنه الأقوى والأسرع, وبشهادة الكثير .. نعم صحيح من خلال تجربتي به كانت جداً رآئعة .. لكن لا تنسى إنه لا يدعم خدمات جوجل بلاي لذلك أعتقد البعض منكم سوف يحتاج الأثنين معاً, أو يستطيع استخدام الجيني موشن من أجل معاينة التطبيقات والجوال الحقيقي من أجل التطبيقات التي تتعامل مع خدمات جوجل. وسوف تحتاج المحاكي الافتراضي في حال مثلاً قمت بعمل تشات على سحابة جوجل للتراسل, هكذا تحتاج لأكثر من جهاز

والمحاكي الافتراضي سوف يفي بالغرض لذلك.

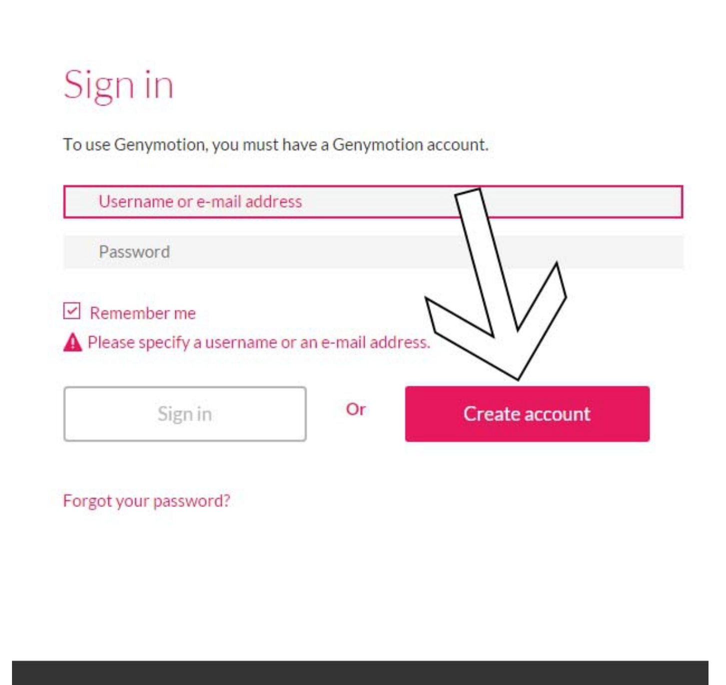
كي لا أطيل الحديث عليك سوف نقوم بالمطلوب, في البداية يجب علينا الذهاب إلى موقع الجيني موشن وفتح حساب مجاني لديه.

للدخول إلى موقع الجيني موشن : https://www.genymotion.com

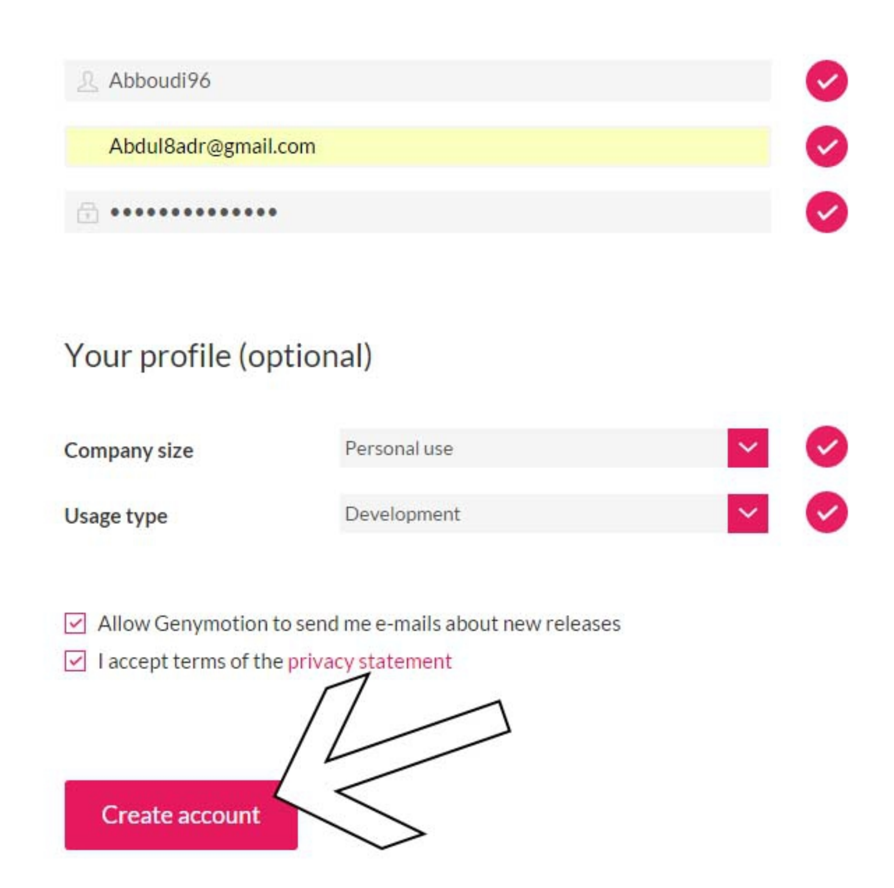
اضغط على زر Sign in



ثم أضغط على إنشاء حساب Create account

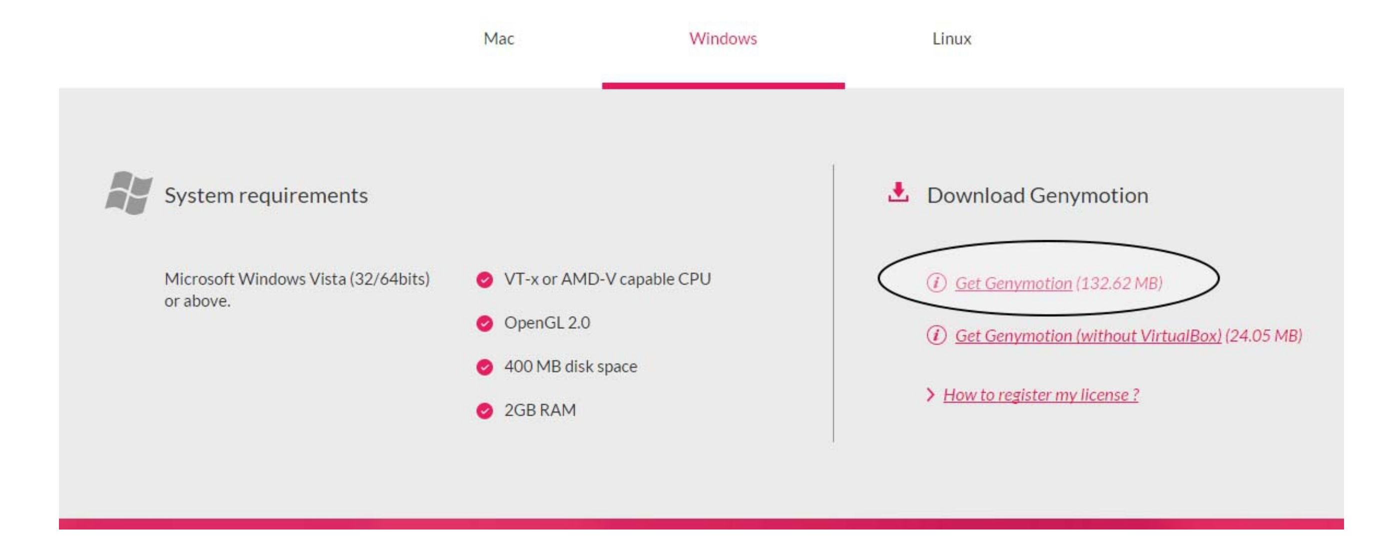


قم بملأ المعلومات الخاصة بك للتسجيل.



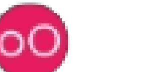
بعد الانتهاء من البيانات, اضغط على Create account

بعد تسجيل الحساب, ندخل إلى صفحة التحميل https://www.genymotion.com/#!/download



نختار نظام جهازنا, ونضغط على تحميل النسخة الكاملة.

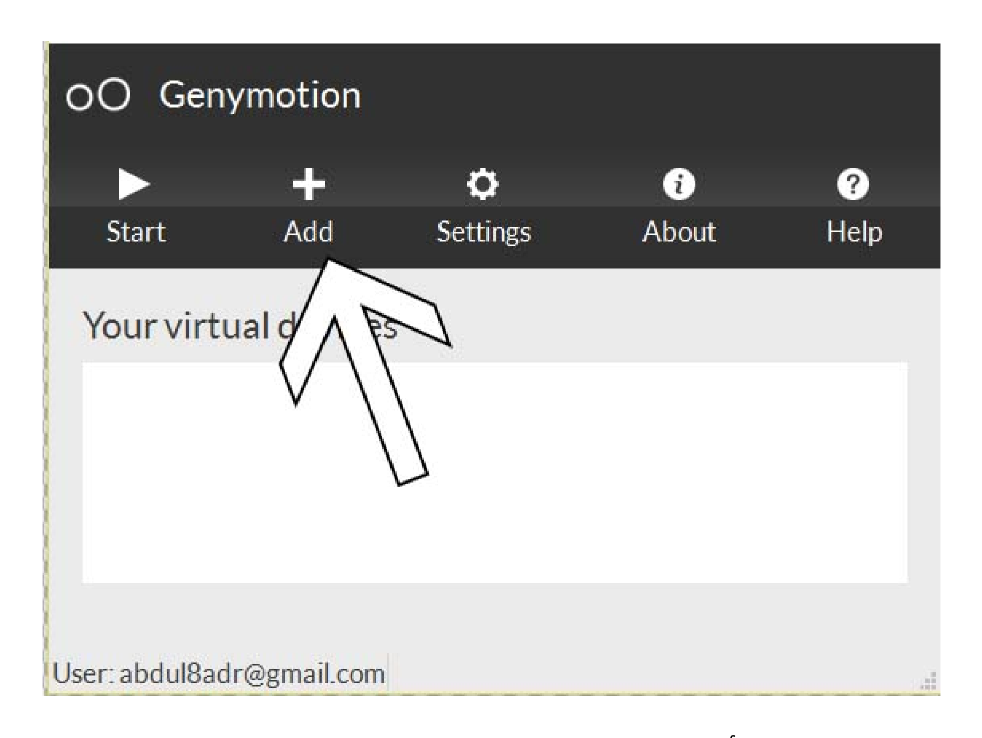
بعد تحميل الجيني موشن, نقوم بتثبيته على جهازنا كاي برنامج آخر. بعد التثبيت نقوم بفتح برنامج الجيني موشن



بعد فتح البرنامج سوف يطلب منك المعلومات التي سجلت بها في الموقع, الإيميل وكلمة المرور.

بعد الأدخال يخبرك إنه لا يوجد لديك إي جهاز هل تريد إضافة جهاز ؟ تقول له نعم.

أو قم بالضغط على اضافة Add

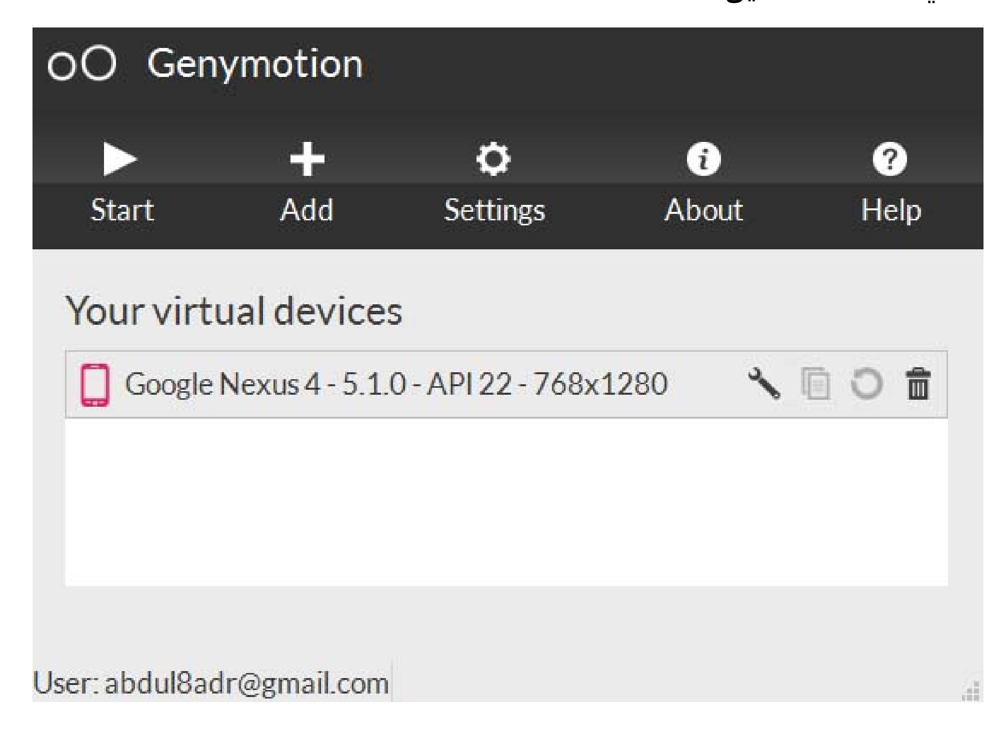


سوف تظهر لك نافذة فيها عدة أجهزة, اختر منها ما يناسبك, حسب المستوى API او الحجم والدقة . ويوجد اجهزة سامسونج وسوني و htc وغيرها.

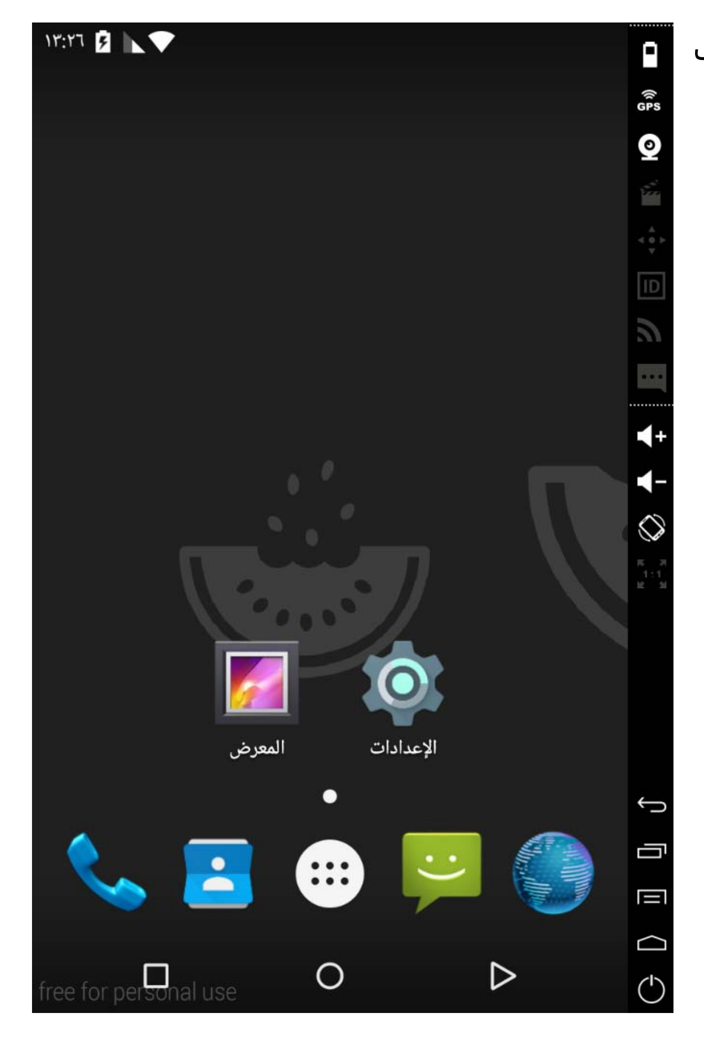
Select a new virtual device
Android version: All Device model: All
Available virtual devices
Google Nexus 4 - 5.1.0 - API 22 - 768x1280
!! PREVIEW - Google Nexus 5X - 6.0.0 - API 23 - 1080x1920
!! PREVIEW - Google Nexus 6P - 6.0.0 - API 23 - 1440x2560
Custom Phone - 4.1.1 - API 16 - 768x1280
Custom Phone - 4.2.2 - API 17 - 768x1280
Cancel Next

بعد الاختيار اضغط التالي Next.

اضغط كذلك التالي .. سوف يتم تنزيل الملفات الخاصة بهذا المحاكي من الأنترنت. وبعد الانتهاء سوف يظهر لك في نافذة التشغيل :

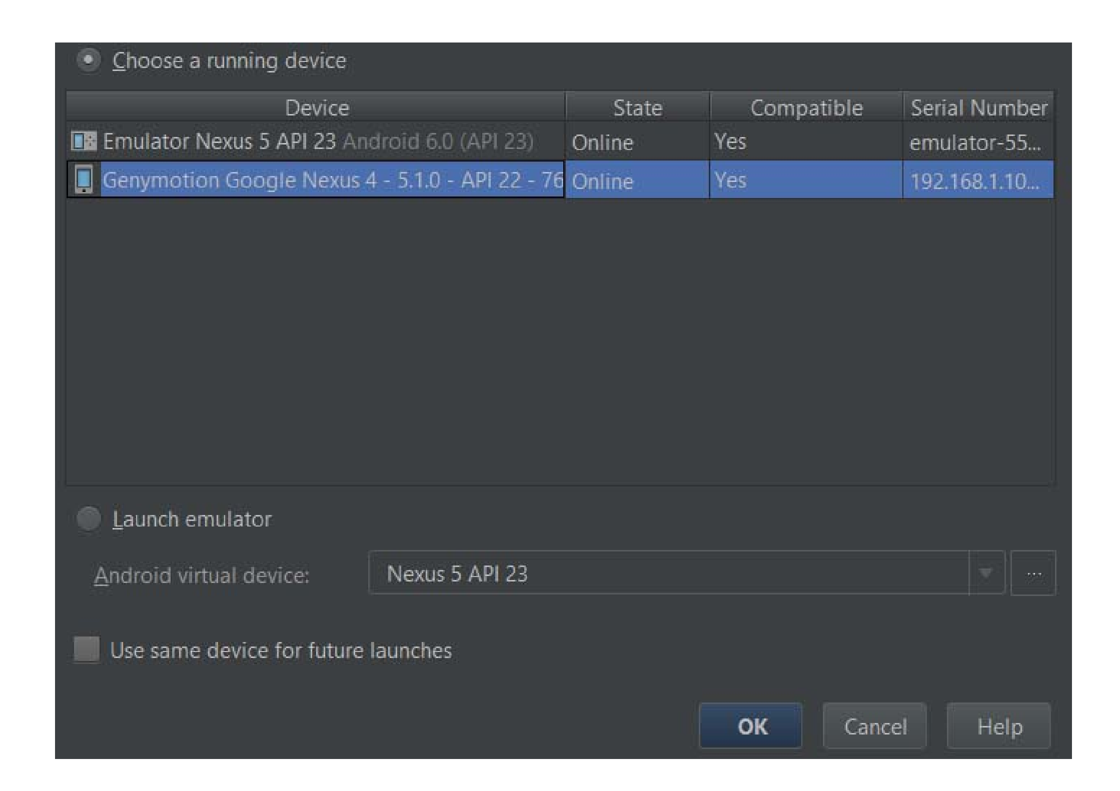


وإن واجهت مشاكل, من خيار Settings ثم ADB إن متصل بحزمة SDK الافتراضية, فإن كانت لديك في جهازك مختلفة المسار قم بتحديدها بشكل يدوي.



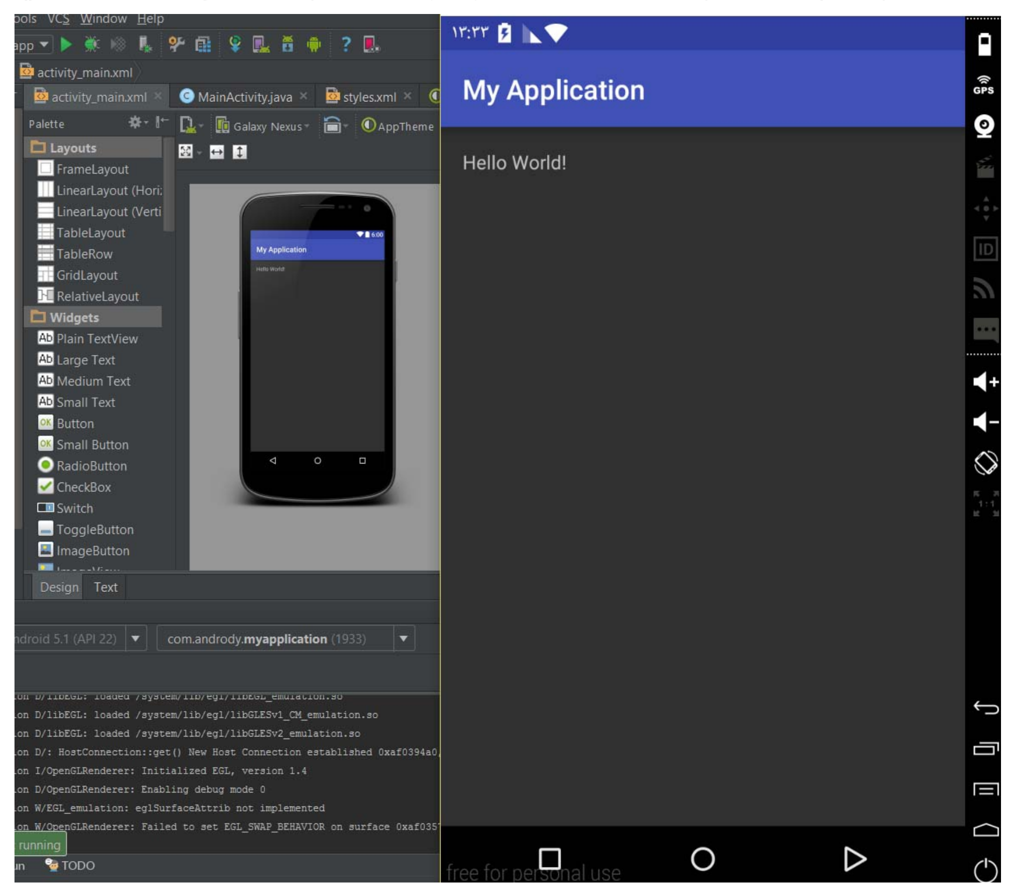
و الآن قم بالضغط على زر Start لبدأ تشغيل المحاكي .... والنتيجة : >>

لاحظ الآن عند الذهاب لمشروعنا في الأندرويد ستوديو والضغط على زر تشغيل المشروع سوف تظهر نافذة التشغيل وفيها جهازين ..



الجهاز الأول المحاكي الافتراضي .. والثاني محاكي الجيني موشن ..

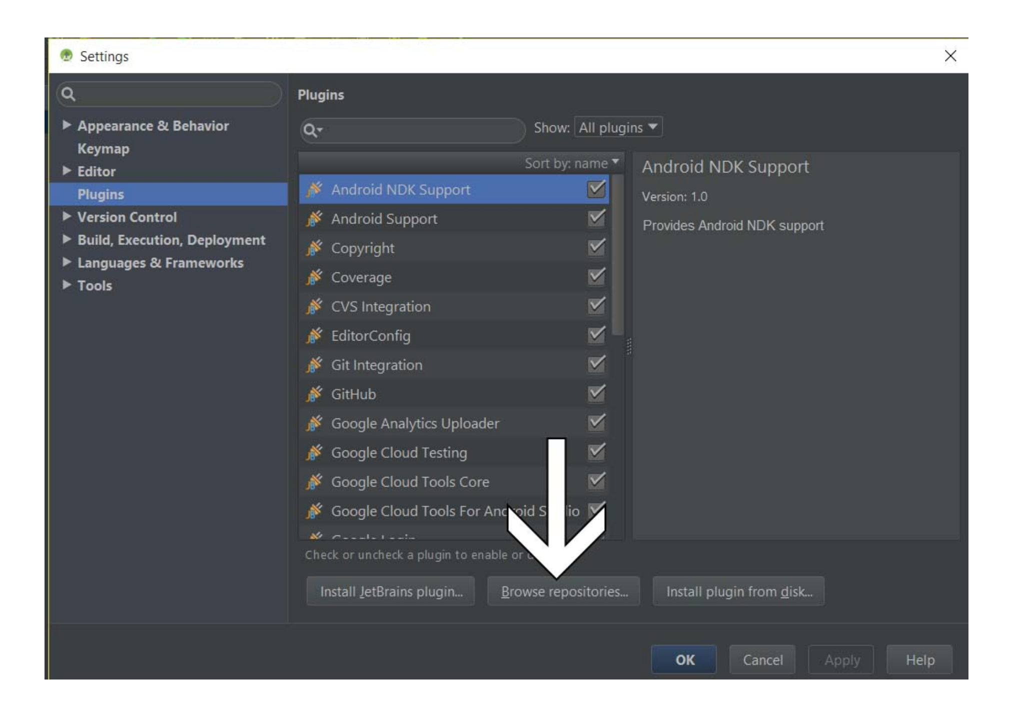
وعرضت لك الاثنين لأنهما الاثنين يعملان الآن .. وتجربة تشغيل التطبيق على الجيني موشن :



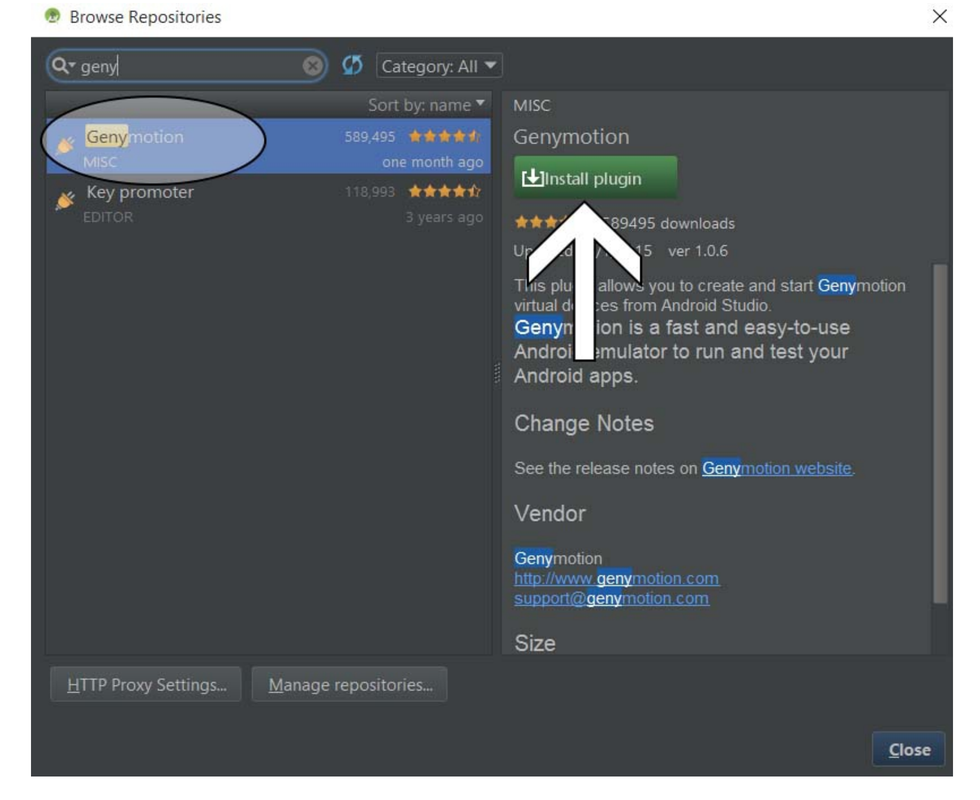
والآن سوف نجدها متعبة إنه يجب علينا تشغيل برنامج الجيني موشن من ثم تشغيل المحاكي ..

لذلك سوف نقوم بتنصيب إضافة لبرنامج الأندرويد لنستطيع من خلاله تشغيل الجيني موشن بشكل أسرع.

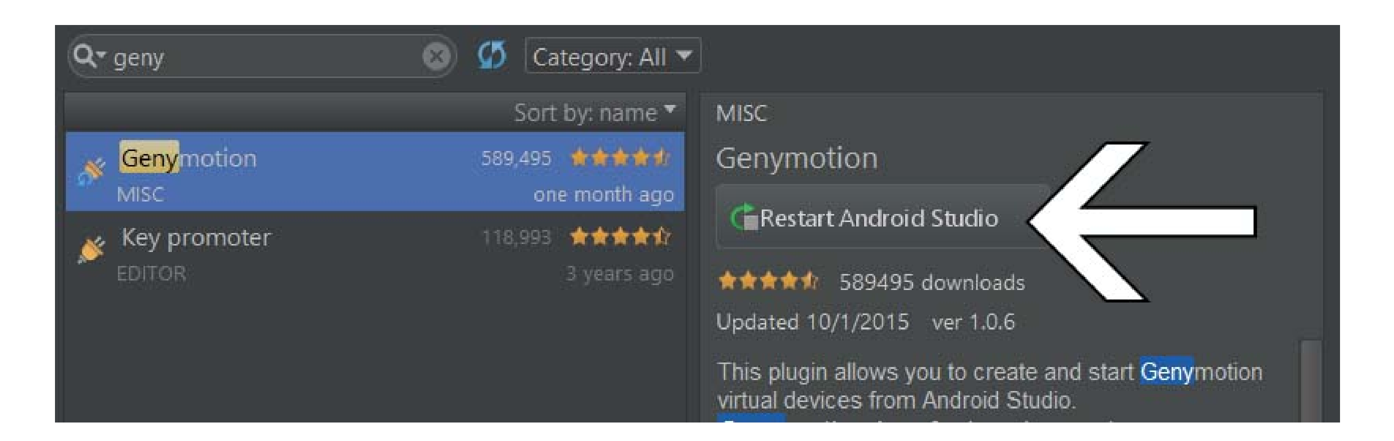
من برنامج الأندرويد ستوديو إضغط على File ثم اختر Settings, سوف تظهر لك نافذة الإعدادات, اختر منها الإضافات Plugins .



نضغط على زر Browse repositories, لتخرج لنا نافذة جديدة ونكتب داخلها في شريط البحث geny

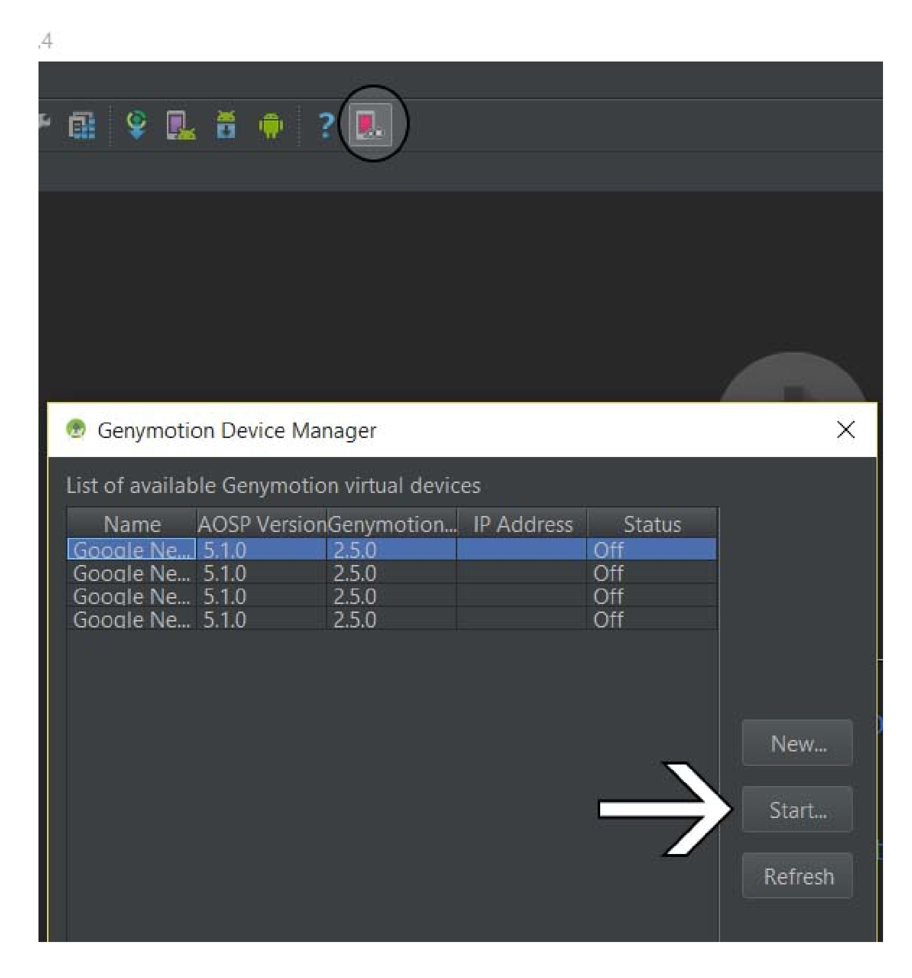


من ثم نختار الجيني موشن Genymotion ونضغط على Install plugin .



بعد التحميل اضغط على Restart Android Studio ليقوم الأندرويد استوديو بإعادة تشغيل نفسه.

وبعد التشغيل من جديد سوف تجد في شريط الأدوات شعار الجيني موشن, اضغط عليه لتفتح لك نافذة صغيرة فيها الاجهزة التي إنشئتها على الجيني موشن, اضغط على الجهاز الذي تريده واضغط زر التشغيل Start كما في الصورة :



وهكذا تستطيع إضافة وتشغيل الأجهزة من الأندرويد ستوديو بسهولة.

معاينة مشروعك الأندرويد على جوالك

بعد ما تعلمنا كيفية معاينة مشروعنا الأندرويد على المحاكي الافتراضي ومحاكي الجيني موشن , في هذا الدرس سوف نتعلم كيفية معاينة التطبيق على جهازنا أو جوالنا الحقيقي.

في البداية يجب علينا القيام بتفعيل تصحيح usb أو **USB debugging** من إعدادات الجوال, نذهب إلى الضبط – خيارات المطور , من ثم نقوم بتفعيل تصحيح USB .

وفي حال لم تجد خيارات المطور في جهازك, إذهب إلى الضبط – حول الهاتف – رقم الإصدار, قم بالضغط عليه سبع مرات متتالية .. ثم أرجع للخلف تجد خيارات المطور قد ظهرت لك.

بعد تفعيل خيارات المطور وتفعيل تصحيح يو إس بي. قم بتوصيل جهازك بالكمبيوتر (يجب أن يتعرف الحاسوب على جهازك إي التعاريف الخاصة بجهازك موجودة بالحاسوب).

ومن ثم إنقر زر التشغيل سوف تجد جهازك في نافذة التشغيل قم باختياره وسوف يتم تشغيل التطبيق على جوالك. يوجد جوالات عليها حماية تطلب تصريح لوصول الحاسوب للجهاز وافق عليها.

وإن واجهتك مشاكل جرب إن تضع الأمر التالي في ملف AndroidManifest.xml :

داخل أوسمة الإبليكيشن :

```
android:debuggable="true"
```

وإن شاء الله سوف يعمل معك بشكل سليم .

إذا كان جهازك من شركة جوجل, أجهزة Nexus, لتعريفها على جهازك, من حزمة الـ SDK قم بتنزيل Google USB Driver.

أنواع البيانات

أنواع البيانات التي سوف نتعامل بها مع الذاكرة في الحاسوب, تختلف أسمائها في بعض اللغات لكن الهدف واحد, و يجب معرفتها لكل مبتدئ فسوف يتم استخدامها بكثرة أثناء برمجة التطبيقات.

من الأفضل لك أن تبحث عنها بشكل موسع, ولكن سوف أذكر هنا بشكل مختصر ما نحتاجه مبدئياً.

النوع	مثال	--
int	1000	عدد صحيح
double	5.2	عدد عشري
char	A	حرف
Boolean	True or False	قيمة منطقية

والنصوص هي من نوع String هي مجموعة أو مصفوفة من char لذلك لا تعد من الأنواع الأساسية. ولكن هي الأكثر استخداماً مثال :

String **AnyName** = “**Abboudi_Aliwi**”

كذلك أخي القارئ أحب إن أوضح إن الإعداد الصحيحة أنواع من حيث القيم وتختلف ..

مثلاً byte تستطيع التخزين داخله من -128 إلى 127, ويوجد أنواع أخرى وذكرت في الجدول نوع int لأنه يستطيع تخزين قيمة كبيرة ويوجد أكبر منه, ولكن هو الأكثر انتشار أو استخدام.

فالـ int يستطيع تخزين ما بين -2147483648 إلى 2147483647 فهي قيمة كبيرة جداً لذلك تستطيع استخدامه بكل سهولة للأعداد الصحيحة ويوجد أكبر منه long.

هي أنواع كثيرة .. وكلاً منها له قيمة أو سعة تخزينية, أبحث عن الموضوع بنفسك .. ولكن حالياً المهم لك أن تعرف ما بداخل الجدول .. في حال قلت لك استخدم سترنج إي نص, أو انت - انتجر اي عدد صحيح وهكذا. وسوف تتكرر معناً كثيرة كلمة سترنج, لأنها له استخدام في الذاكرة بالجافا, ولها استخدام مختلف بالأندرويد بشكل عام وكذلك بداخل ملفات xml.

وقيمة Boolean أكثر من مهمة, مثلاً نبرمج زر يفحص هذه القيمة إن كانت ترو نفذ الأمر التالي, وإن كانت فولس نفذ أمر مختلف, و أزرار أخرى تتحكم بهذه القيمة. وبذلك تستطيع عمل اختبارات والكثير فقط من هذه القيمة الأكثر من رائعة.

المتغيرات والقيم

ما كان يجب عليي أن أدخل أكثر في الجافا, ولكن معلومات يجب عليك معرفتها لكي تستطيع برمجة تطبيقات الأندرويد دون إي مشاكل إن شاء الله.

في الجافا دائماً ما نذكر النوع ثم المتغير ثم قيمة هذا المتغير, وهذا الشيء أعتقد معروف في عدة لغات ليس فقط الجافا.

و السبب الذي أريد ذكره هنا, الكيفية + التسمية.

أمثلة لتفهم عن ماذا أتكلم بشكل أوضح :

```
String myname = "Abboudi Aliwi";
```

النوع هو **String** و المتغير هو **myname** والقيمة **Abboudi Aliwi**

ما نلاحظه : إن النوع هي أنواع معدودة وقد ذكرنا بعضها في الدرس السابق أنواع البيانات, و المتغير له قواعد في اختياره, إما القيمة فهي تبعاً للنوع وما دامت سترنج إي يجب وضعها داخل "" بالإضافة إلى لا مشكلة بوجود الفراغات وغيرها من الحروف والأرقام.

اختيار أسم المتغير :

- لا يجب إن يبتدأ برقم, لا مشكلة يبتدأ بحرف أو علامة _ أو $.

- لا مشكلة في كتابة الأرقام في وسط او نهاية المتغير.

- لا تستخدم في اسماء المتغيرات العلامات عدا _ أو $.

- المتغيرات تختلف وتتأثر بحالة الأحرف الكبيرة والصغيرة.

مثال : int Abboudi هو مختلف تماماً عن متغير int abboudi.

- الأفضل الابتعاد عن المتغيرات من الحرف الواحد.

- ابتعد عن الكلمات التالية في اسماء المتغيرات : (كلمات اساسية للجافا)

abstract - assert - boolean - break - byte - case - catch - char - class - const - default - do - double - else - enum - extends - final - finally - float - for - goto - -if - implements - import -

instanceof - int - long - native - new - package - private - protected - public - return - short - static - strictfp - super - switch - synchronized - this - throw - throws - transient - try - void - volatile - while - false - null - true

* * عندما تريد تغيير قيمة أحد المتغيرات التي تم تعريفها سابقاً او التأثير عليها, لا تكرر كتابة النوع, فقط المتغير وقيمته الجديدة.

مثال للتعرف على المتغيرات من داخل الأندرويد ستوديو :

```
// متغيرات وقيم صحيحة
    byte Aa = 100;
    int Bb = 1000;
    double Cc = 5;
    double Dd = 2.5;
    String Ee = "Hello my friends";
    boolean Ff = true;
    boolean Gg = false;
    char Hh = 'H';

// متغيرات وقيم خاطئة
    byte AAA = 1000;

    int new = 10;

    double for = " !! ";

    String BBB = 10;

    boolean CCC = "true";
```

والآن يجب عليك قراءة أكثر عن المتغيرات في الجافا .. وسوف نتعرف عليها أثناء العمل بالمشاريع البسيطة ..

** عند وجود متغير كما في الصورة السابقة بلون مختلف - اللون الرمادي - هذا يعني إن المتغير هذا وجوده كعدمه إي غير مستخدم في هذا المشروع. وعند استخدامك له سوف يتغير لونه.

أهم العناصر داخل اللياوت

الآن نبدأ بالدروس التي منها سوف نطور مشروعنا الأول, وهنا سوف نأخذ أهم العناصر والتي تتكرر وموجود في جميع التطبيق أو أغلبها. ومهم جداً إن تحفظ اسمها برمجياً.

العنصر	الاسم البرمجي
النص	TextView
الزر	Button
الصورة	ImageView

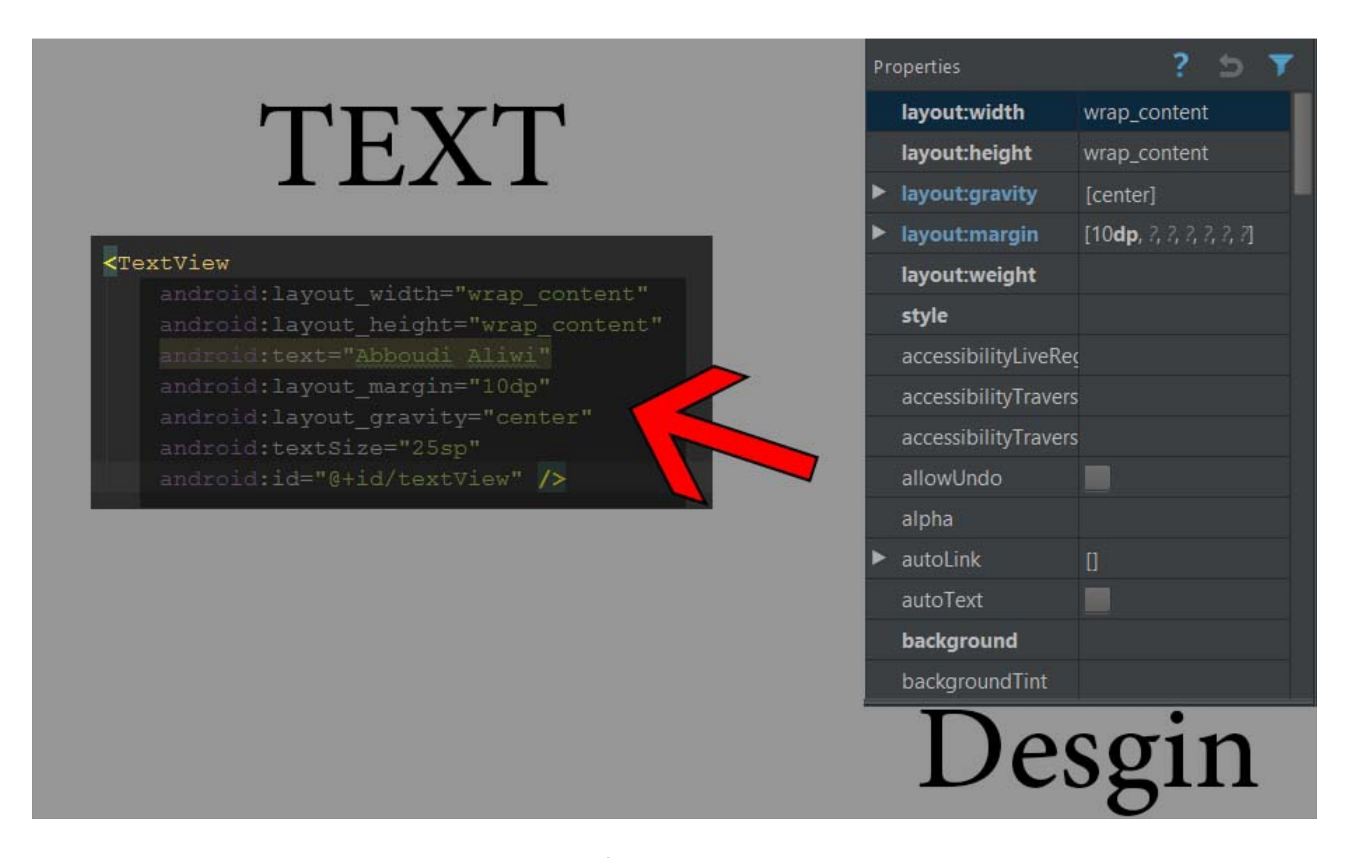
لا شك إنه لا يخلو إي تطبيق من النصوص, لذلك فالنص أو التكست فيو يعتبر العنصر الأهم من العناصر في الواجهات التي تعرض لمستخدم التطبيق. فتستطيع من خلالها توضيح الإشارات والدلالة إلى إشياء معينة, وأما الزر فبواسطته تستطيع التنقل بين أرجاء التطبيق, ولا يخلو تطبيق من أزرار, وأما الصور فهي تزيد جمالية وتحسين منظر التطبيق.

في الصورة التالي تستطيع معرفة العناصر هذه بشكل اوضح :



شرح أهم خصائص العناصر

في البداية أصدقائي ما أقصده بالخصائص هو :



هي الخائص لكل عنصر, من شاشة التكست فيو تظهر أسفل العنصر جميع الخصائص المستخدمة له, اما من شاشة الديزاين تظهر لك جميع الخصائص وتستخدم ما تحتاج منها. وتتغير بعض الخصائص على حسب العنصر الذي تم تحديده.

الخصائص المشتركة لجميع العناصر ولا تستطيع الاستغناء عنا هي :

عرض العنصر **android:layout_width=""**

ارتفاع العنصر **android:layout_height=""**

و يوجد لبعض العناصر خصائص خاصة به كذلك اجبارية له. تختلف حسب العنصر.

و الخاصية التي تجب استخدامها لكل عنصر تريد التحكم به برمجياً هي خاصية المعرف.

معرف للعنصر **android:id=""**

وسوف أذكر الآن الخصائص وشرحها والقيمة التي تحملها.

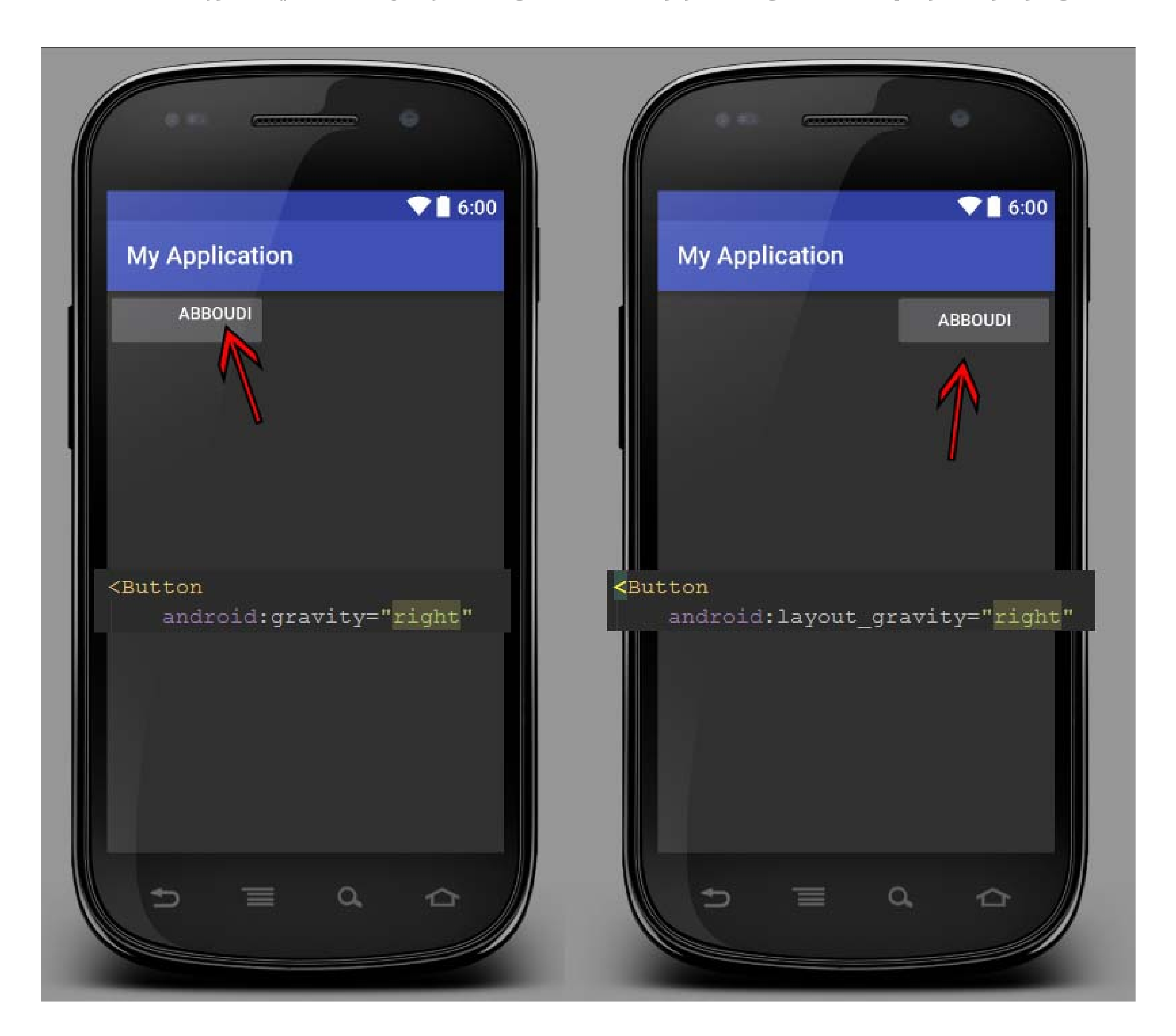
قبل إن ابدا بشرح أهم الخصائص, أغلب الخصائص بهذا الشكل :

android:layout_gravity="center"

تبدأ بـ لياوت, ولكن بعضها تستطيع استخدامها بدون لياوت :

android:gravity="center"

والفرق هو, إن لكل عنصر حد, مثلاً حدود للنص وما بداخل الحدود .. ما كلمة لياوت نقصد العنصر بالكامل, وبدون لياوت إي ما بداخل العنصر وغالباً ما بداخل العنصر نص. مثال في الصورة :



يعني لو مثلاً قلنا لياوت العنصر إي العنصر كامل نقصده, وكلمة لياوت نقصد فيها الملف كامل, وكذلك نقصد فيها الأب الأول لهذا الملف أو أول عنصر بهذا الملف الذي يحتوي داخله جميع العناصر.

خاصية العرض والارتفاع :

android:layout_gravity=""

android:layout_width=""

هذه الخاصيات كما قلت, إنها أساسية لجميع العناصر دون استثناء, فالعنصر الذي لا يحتوي قيمة لعرضه او ارتفاعه يساوي صفراً إي لا شيء, وهي تأخذ ثلاث قيم بالإضافة إلى عدد.

بالنسبة للأعداد لن أتطرق إلى الوحدات والفرق بينها تستطيع مراجعتها من مدونة اندرودي عربي , ولم أذكرها هنا لأنها ليست مهمة كثيراً ما يهم أن تعرفه فقط عند وضع قيمة عددية لإي عنصر استخدام بعد العدد الوحدة dp وإن كانت الوحدة من أجل حجم النص استخدم sp.

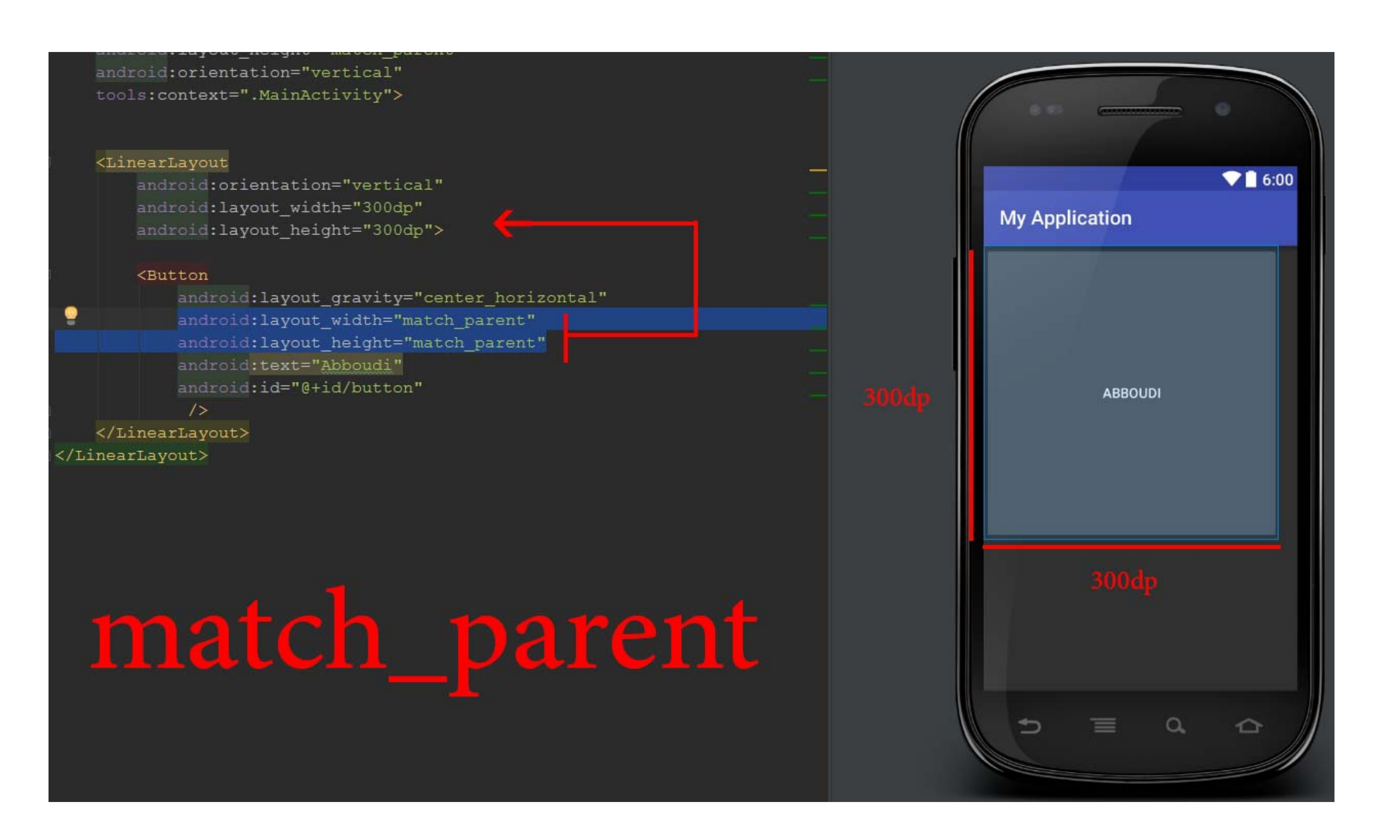
تأخذ داخله القيم : wrap_content, match_parent, fill_parent , وتوضع القيمة هذه بداخل "".

بالنسبة لقيمة fill_parent تم إهمالها بعد المستوى API 8 و أصبحت هي match_parent.

match_parent / يأخذ العنصر العرض أو الارتفاع حسب العنصر الأب له.

wrap_content / يأخذ العنصر العرض أو الارتفاع الذي يحتاجه فقط.

هذه الصور أوضح فيه لكم الفرق :







بالنسبة لجميع القيم السابقة كلها تتأثر بالعنصر أو اللياوت الأب لهذا العنصر, لأننا مثلاً لو وضعنا wrap_content للعنصر إي يأخذ ما يحتاجه, ولكن الأب لهذا العنصر جداً صغير ليس لديه كافي ليعطيه حجمه لذلك يصبح أصغر, ولا ننسى إن الوحدة dp تتأثر بحسب المساحة أو قياس الشاشة المتاح.

مع الاستخدام سوف تتعلم كيف تفرق بينها ولا يمكن للعنصر إن يخرج عن حدود الأب له.

خاصية الاتجاه :

android:orientation=""

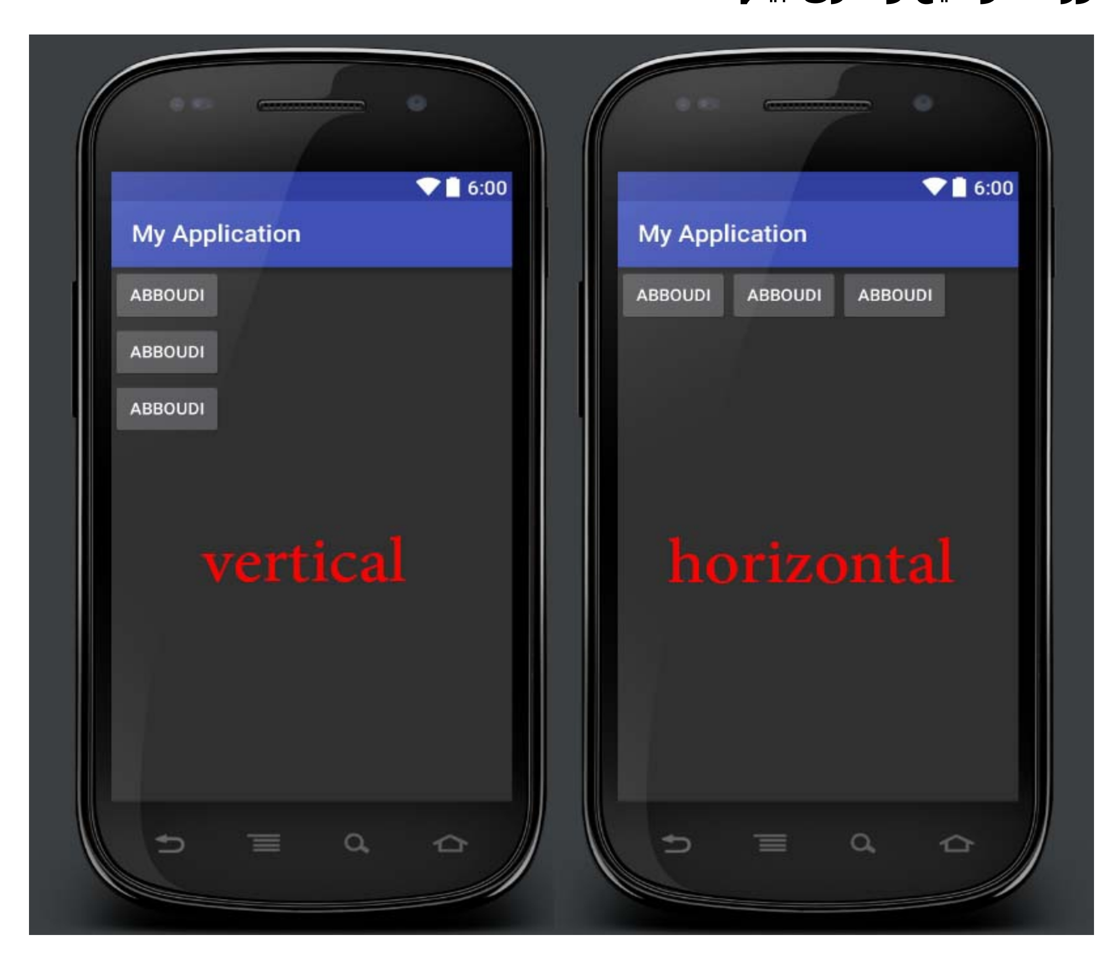
وهي الخاصية التي تعتبر اجبارية لـ LinearLayout إي النمط أو الترتيب الخطي, القيمة الافتراضية له هي horizontal وجميع الخصائص الاخرى لها قيم افتراضية ونقصد بهذا إن في الحقيقة جميع الخصائص مفعلة او موجودة للعنصر ولكن غير ظاهرة لأنها تأخذ القيمة الافتراضية فلو استخدمتها إذا أنت تريد تغيير القيمة الافتراضية لها .

وهي غير مهم وجودها إذا كانت اللياوت LinearLayout لا تحتوى سوا عنصر واحد, لأنها سوف تأخذ كما ذكرت لك الترتيب الأفقي, ولكن لو كانت عناصر كثيرة يجب استخدامها وتحدد بداخلها أحد القيمتين : horizontal أو vertical.

horizontal / الترتيب الأفقي للعناصر إي بجانب بعضها.

vertical / الترتيب العمودي للعناصر إي فوق بعضها.

وهذه صورة للتوضيح والفرق بينها :

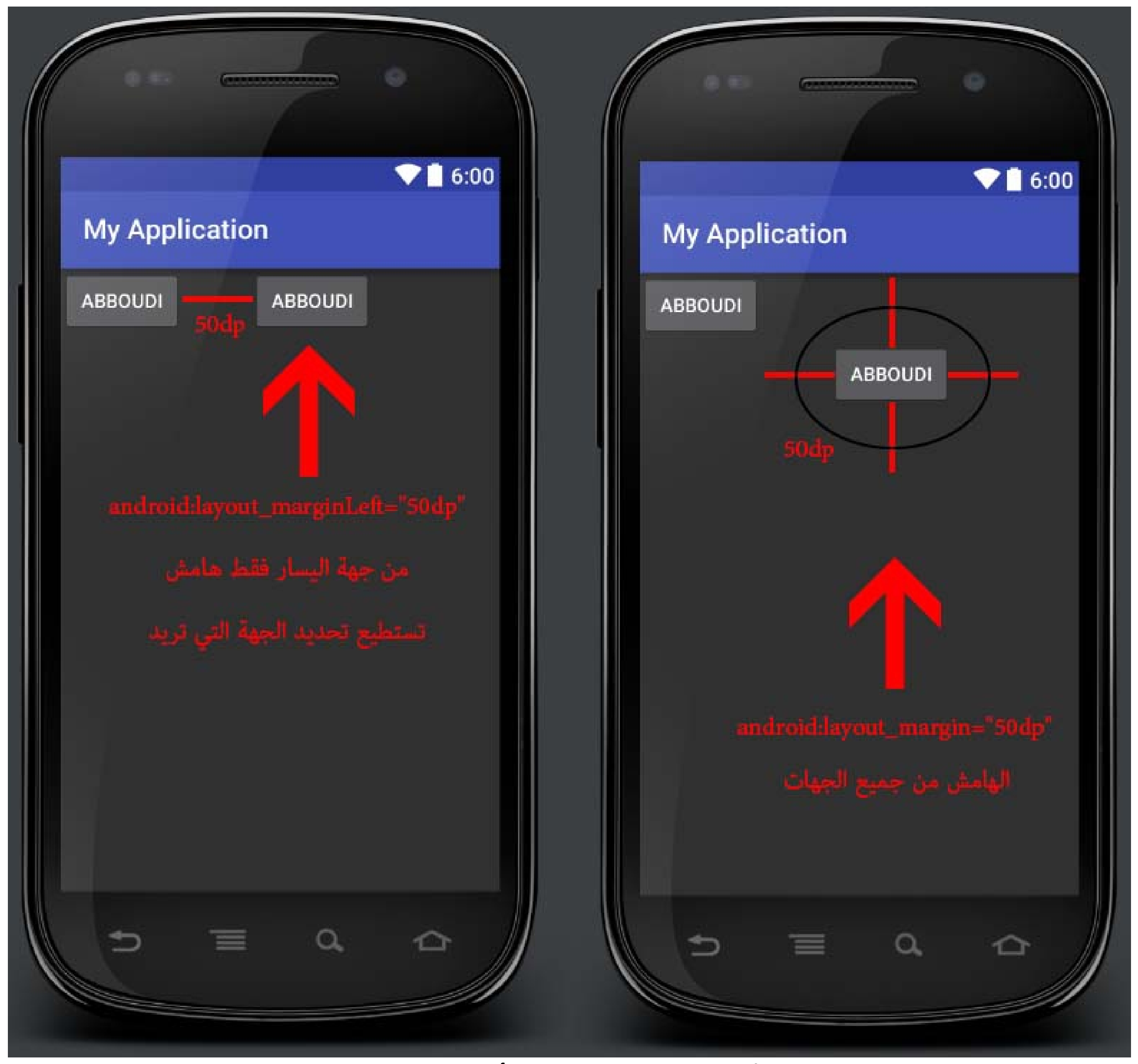


خاصية الهامش :

android:layout_margin=""

الهامش خاصية مميزة وعن نفسي لا أستطيع الاستغناء عنها في إي تطبيق أو عنصر, لأنها جداً مهمة لتنسيق التطبيق, فهي هوامش العنصر التي ليست من حدوده, بشكل أوضح إنها المسافة ما بينها وما بين ما حولها, وتستطيع استخدامها لتحديد الهامش من جميع الجهات أو من جهة معينة.

هذه الصورة توضح لكم استخدامها :



** بالنسبة للجهات في برمجة الأندرويد سواء كان java أو xml. لكن في الجافا الاحرف كبيرة وفي xml الاحرف صغيرة.

top / الأعلى   bottom / الأسفل

right / اليمين   left / اليسار

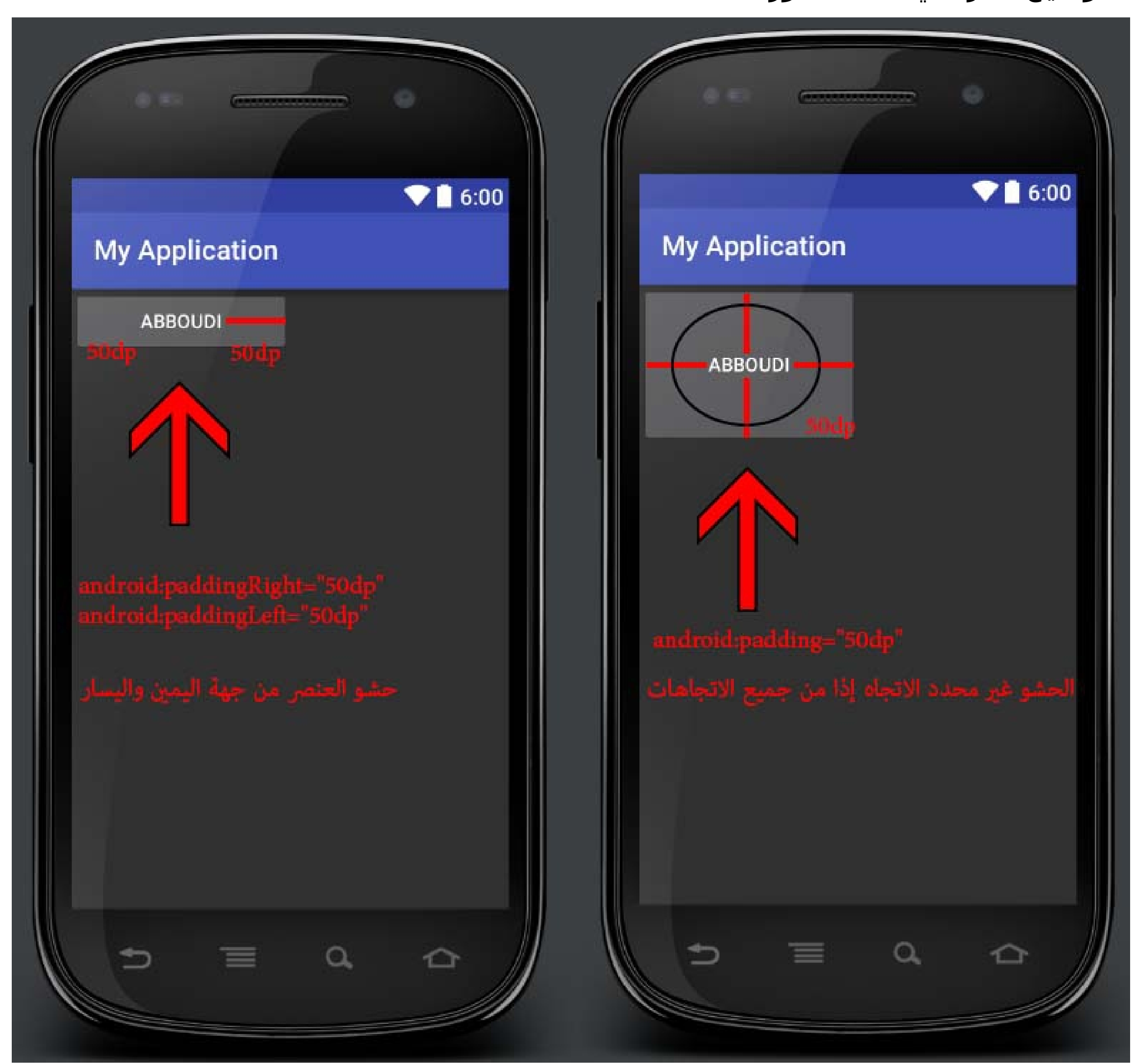
خاصية الحشو :

android:padding=""

الخاصية هذه مفيدة كذلك, وهي تقريباً تشبه خاصية الهامش ولكن بشكل عكسي, فهي الهامش ولكن ليس بين العنصر وما حوله, ما بين أطراف العنصر وما داخله. والفرق إن المساحة التي يحجزها الهامش لا تعتبر من ضمن العنصر, إما الحشو فهي من ضمن العنصر, فلو مثلاً لدينا نص, اضغط هنا والحشو الخاص به جداً كبير, وليس ظاهر لأن النص ليس له حدود, لكن مساحة الحشو تعتبر منه لو ضغطنا على إي مكان فيها فسوف ينفذ الأمر الذي تم اعطائه للنص.

كذلك تستطيع إعطاء الحشو للعنصر من إي اتجاه تريد أو من جميع الاتجاهات تماماً مثل الهامش.

للتوضيح أكثر في هذه الصورة :

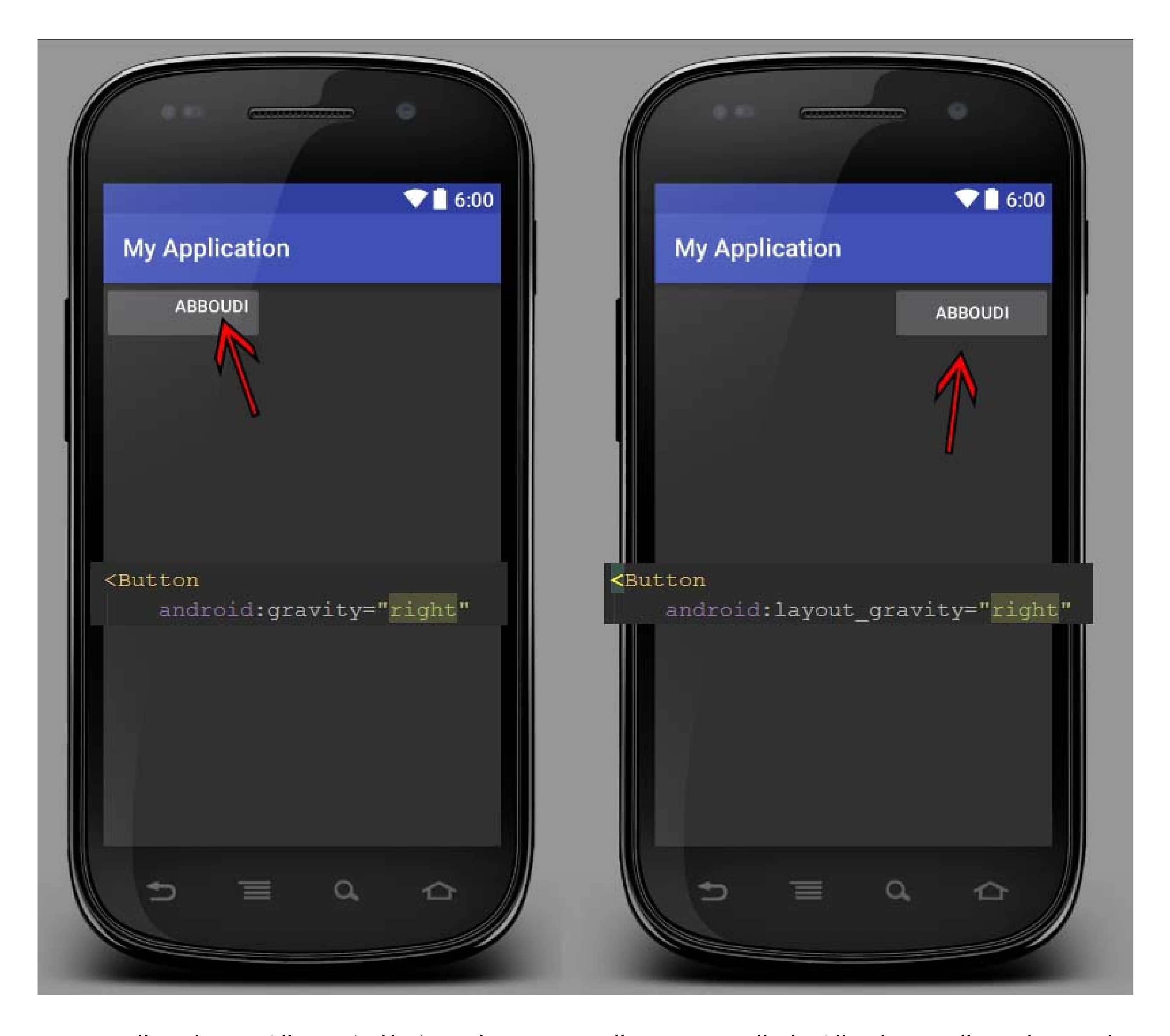


خاصية الجاذبية :

android:gravity=""

android:layout_gravity=""

قد وضحت سابقاً الصورة التالية :



خاصية جاذبية العنصر, إي الاتجاه الذي ينجذب إليه, وهي خاصية كذلك كثيرة الاستخدام, والقيم التي تأخذها هي الاتجاهات, اعلى أسفل يمين يسار, وقيم اخرى.

بالنسبة لـ android:layout_gravity فالعنصر يتحرك بالمساحة الحرة من العنصر الأب له.

أما android:gravity فيتحرك ما بداخل العنصر في المساحة المتاحة له من هذا العنصر.

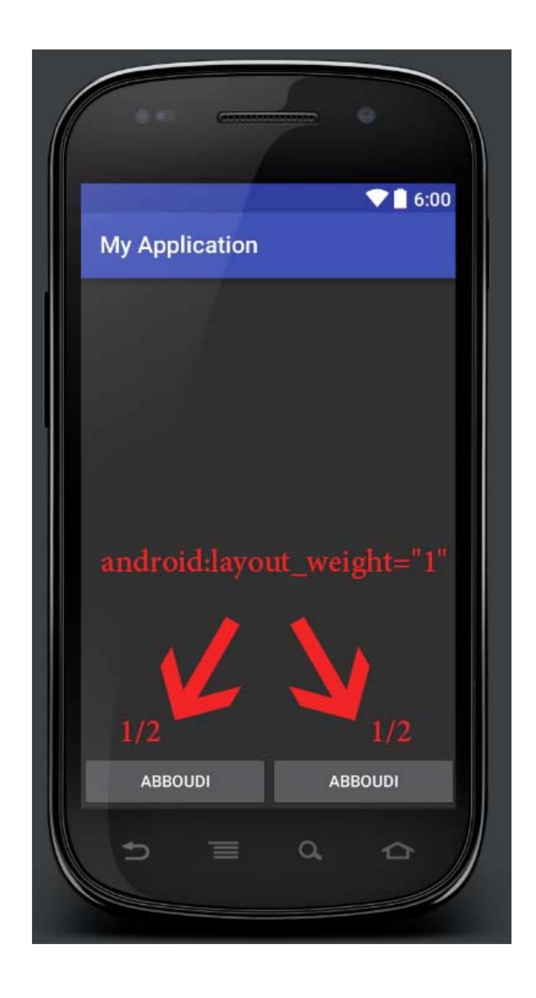
خاصية الوزن :

android:layout_weight=""

أو بما أسميها خاصية التقسيم, لتقسيم الشاشة أو اللياوت للعناصر, لكل منها يأخذ نصيبه. والصور هي فقط تستطيع أن تشرح ما هذه الخاصية, إذا قمت بإعطاء العناصر هذه الخاصية فإستخدم القيمة ما تحت 1 لتتقسم عليهم. مثلاً 0.1 أو 0.5 .. أو من مئة 0.55 أو 0.20 .. أو تحت 10 , مثلاً 1 أو 4 وهكذا. القصد لا تعطي عنصر قيمة 50 وعنصر آخر 0.50.

مثال في الصور :





خاصية المعرف : ID

android:id=""

خاصية المعرف للعنصر, لكي تعطيه إسم يميزه للتعامل معه برمجياً, سواء كان صورة أو نص إلخ وغيرها من العناصر. عن طريق هذا المعرف نستطيع إعطاء الأوامر أو التحكم بهذا العنصر بلغة الجافا.

ويكتب بالشكل التالي :

android:id="@+id/button"

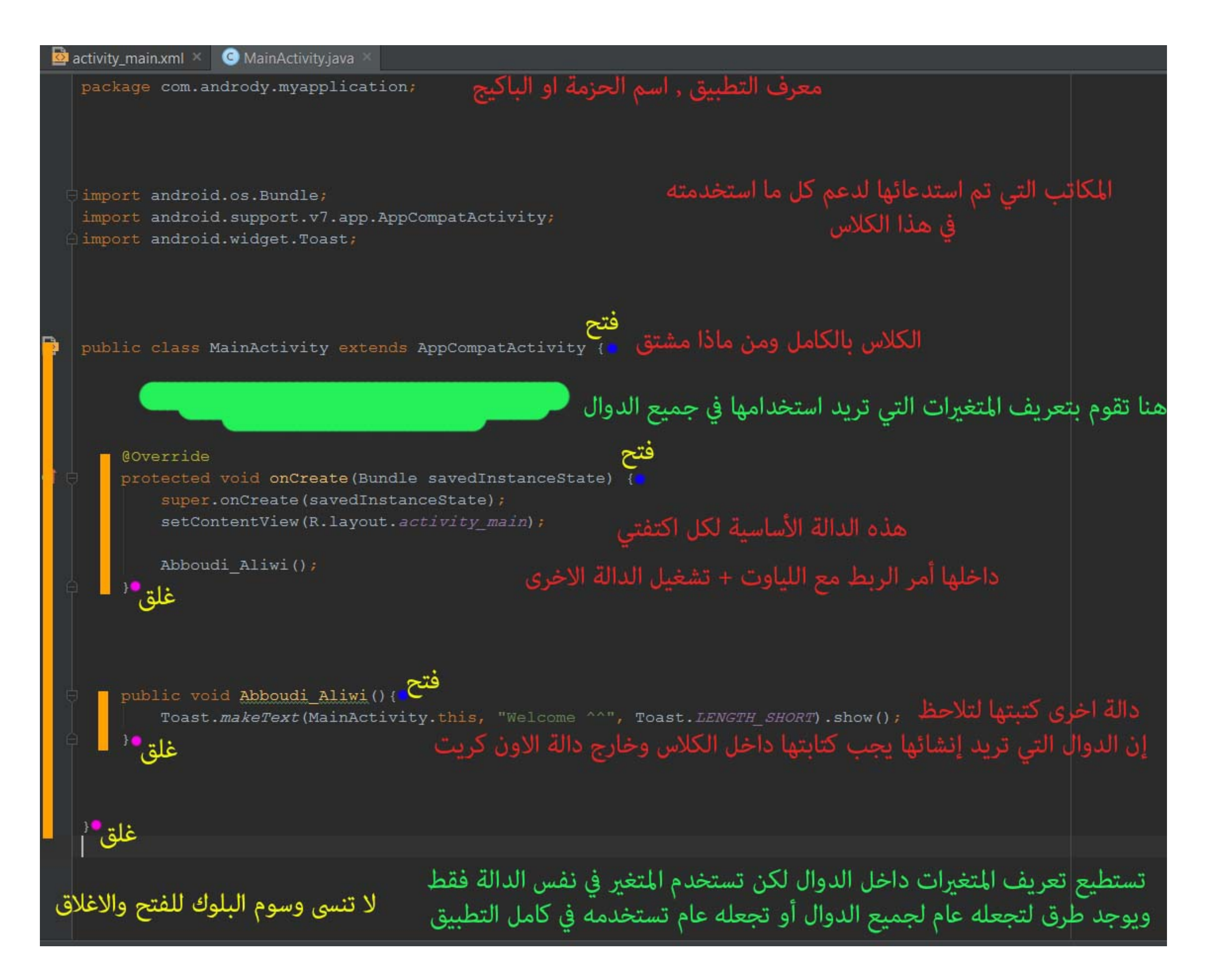
واستبدل button بالذي تريده, ومعنى +id إي اضف هذا المعرف للموارد R, كما إن بدون + يصبح المعنى استخدم المعرف الذي اسمه كذا. وسوف تتعرف عليها اكثر من خلال الاستخدام.

** اغلب الخصائص الاخرى واضحة المعنى النص, لكتابة النص وحجم النص ولونه إلخ. بالنسبة لـ لون النص, تستطيع استخدام الوان معرفة في ملف الالوان بالفايلوس او استخدام الكود للون مباشرة وللألوان أكواد بعدة صيغ تستطيع استخدام الصيغة الأسهل, للتوضيح اكتب داخل textColor القيمة #000 سوف يظهر اللون الأسود للنص, وبجانبه داخل ملفات xml مربع صغير اسود اضغط عليه واختر اللون الذي تريد وتلقائي سوف يتغير الرقم لكود اللون الذي تم اختياره.

التعرف على أهم الأكواد البرمجية وأكثرها تكراراً Java

سوف نتعلم الآن ما أكثر العمليات أو الأكواد أو الدوال التي سوف تتكرر معك في إستخدام الجافا اثناء برمجة تطبيقات الأندرويد. ومهم لك أن تعرفها لأنك سوف تحتاجها وبكثرة. والبعض منها سوف أذكرها لأنك سوف تحتاجها, وأنصحك بمراجعة هذا الدرس أكثر من مرة ليسهل عليك الامر وتحفظه تماماً.

في البداية يجب أن تتعرف على صيغة أو كيفية الكلاس أو الاكتفتي بالأندرويد. شاهد هذه الصورة :



إذا لم يوضح لك إي نقطة في الصورة, لا مشكلة مع البدأ بإنشاء المشاريع سوف نتعرف عليها بشكل أوضح.

لاحظ الترتيب, اسم الباكيج ثم مكاتب الدعم, ثم الكلاس, ثم بداخله المتغيرات والدوال.

أكثر ما سوف يتكرر معك برمجياً هو إضافة المتغيرات, فالمتغيرات اجبارية للتعامل معها برمجياً.

حسب موقع المتغيرات الذي حددته لكم في الصورة, داخل الكلاس وخارج الدوال, يكون تعريفها بكتابة النوع ثم المتغير ثم علامة ; ولا تنسى إن تجعل الأندرويد ستوديو يقوم تلقائياً باستيراد مكاتب الدعم مثل ما تعلمنا سابقاً في الكتاب, ليتم وضع المكتبة الخاصة لهذا النوع.

عموماً التعريف يكون بهذا الشكل, مثال تعريف زر :

Button **Abboudi**;

الأمر جداً بسيط, أخبرنا الكلاس إن لدينا زر واسمه Abboudi ثم ننتقل لمرحلة ربط هذا الزر الذي قمنا بتعريفه مع الزر الموجود داخل اللياوت. بواسطة المعرف الخاص بالزر.

بهذا الشكل :

Abboudi = (Button)findViewById(R.id.button);

ولكن تذكر الربط يكون بداخل دالة الاون كريت التي يتم تنفيذها بعد تشغيل الاكتفتي.

إذا الكود السابق , و الذي سوف أضعه في صورة لكم الآن, هو أكثر كود سوف يتكرر معك لذلك تذكره جيداً مع اختلاف اسم المتغير كما تريد وكذلك النوع حسب الموجود لديك.

```
package com.andrody.myapplication;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    Button Abboudi;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Abboudi = (Button) findViewById(R.id.button);

    }

}
```

لا أنسى في الصورة السابقة, في مرحلة الربط, قلنا له إن المتغير Abboudi هو من نوع Button وأوجد لنا العنصر الذي يحمل المعرف الذي اسمه button وتأكد إن الزر يمتلك المعرف button. وسوف تظهر لك مشاكل في تشغيل المشروع إن قمت بتعريف متغير من نوع نص وربطته بعنصر من نوع زر.

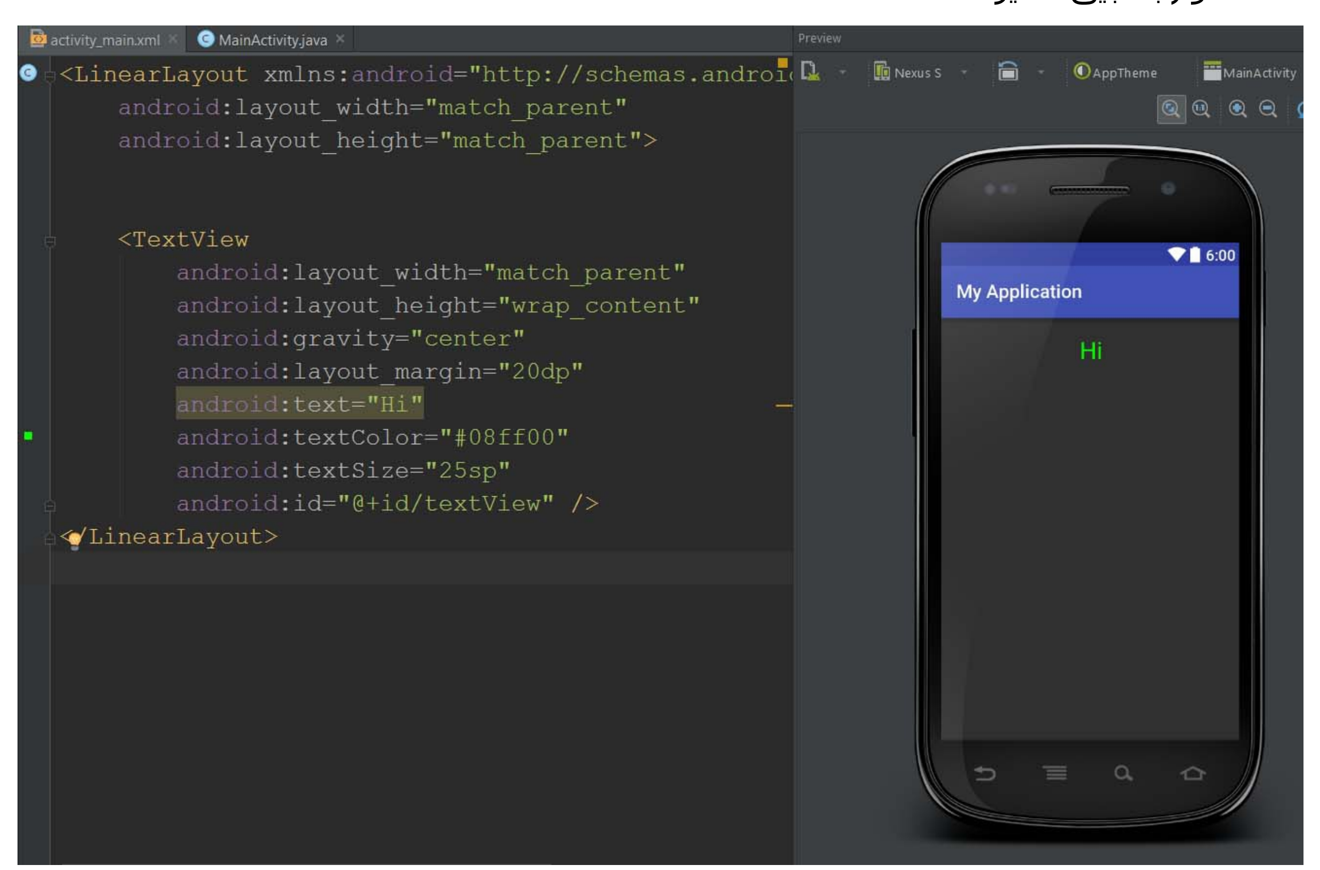
بعد مرحلة التعريف والربط, تأتي الأوامر البرمجية التي تريدها وهي مختلفة. لذلك لن أدخل بها, ولكن سوف نتعلم البعض منها أثناء البدء ببرمجة المشاريع الصغيرة.

الآن سوف نتعرف على الـ get و set .

هذه الكلمتين سوف تستخدمهم بكثيرة والمعنى من get مثلاً أتي لنا بالقيمة الخاص بالعنصر أو بالمتغير أو ,.. وأما set قم بتعديل هذه القيمة.

مثال قم بجلب تاريخ اليوم, وقم بوضع هذا التاريخ داخل نص في التطبيق.

مثلاً لنقوم بتطبيق صغير :



لاحظ هنا الخصائص, العرض له كعرض الأب له, إي كامل الشاشة, والارتفاع حسب حاجته, وموقعه في الوسط, والهامش له 20 و النص Hi واللون فسفوري وحجم النص 25 وأهم خاصية المعرف textView

```
package com.andrody.myapplication;


import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.widget.TextView;

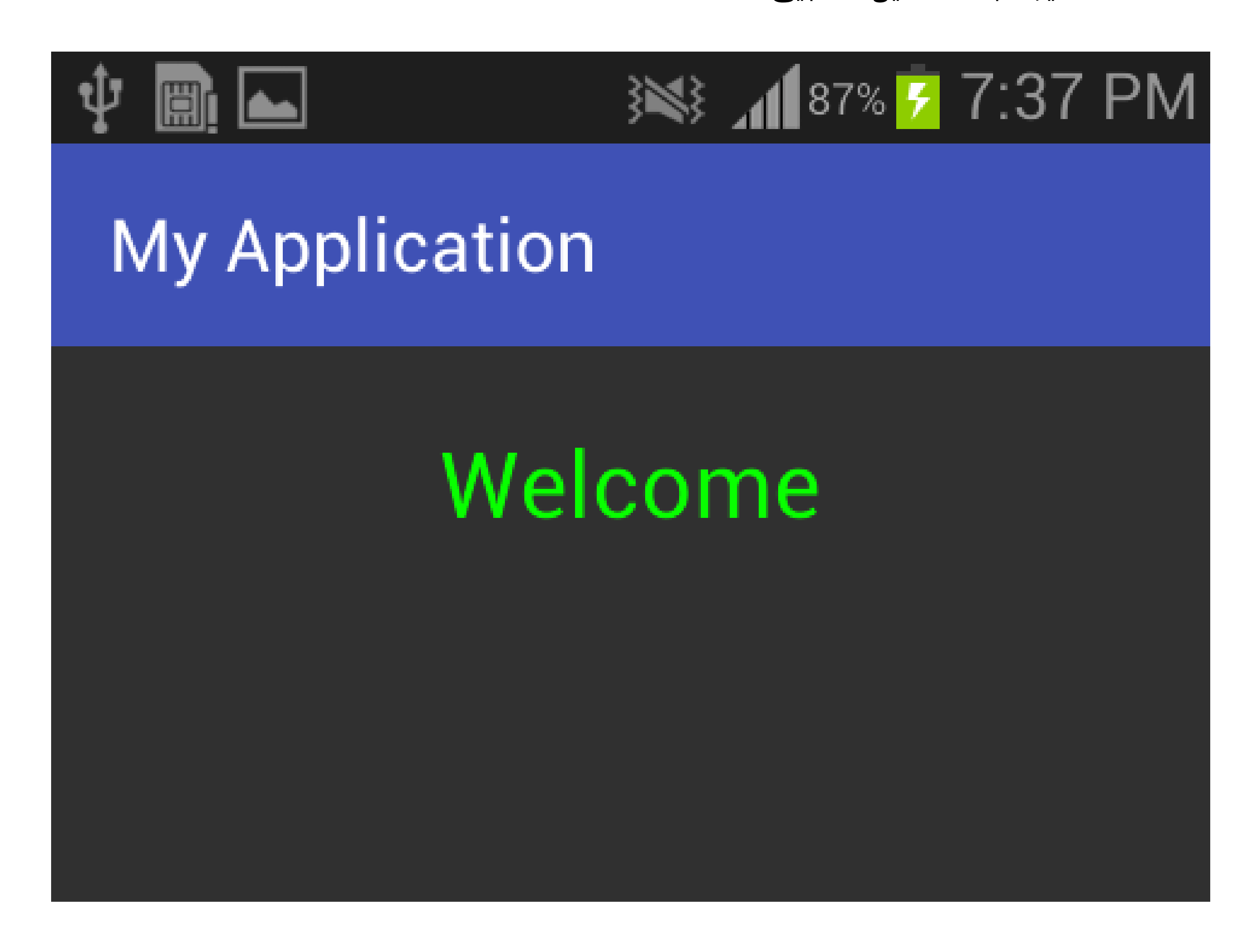

public class MainActivity extends AppCompatActivity {

    TextView Abboudi;       قلنا له يوجد لدينا نص واسمه Abboudi

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Abboudi = (TextView) findViewById(R.id.textView);  قمنا بعملية الربط ما بين النص و بين المتغير
                                  إذا أصبح النص اسمه الجديد هو Abboudi وإي اوامر برمجية نستخدم اسمه الجديد
        Abboudi.setText("Welcome");
       هنا أخبرناه قم بتغير النص بداخل Abboudi وأعطه النص الجديد Welcome
    }


}           لاحظ هذا الأمر لا يتم تنفيذه حتى يتم تشغيل الاكتفتي .. إذا في المعاينة الحقيقة سوف نرى تنفيذ الأوامر
```

الآن لاحظ النتيجة بعد تشغيل التطبيق :



يوجد أمور أخرى كثيرة يجب عليك معرفتها لأنك سوف تستخدمها, ومن أهمها الـ if لوضع شرط, وأكثر من شرط في نفس الوقت واستخدام أي أو و .. برمجياً وجواب الشرط وكذلك كيفية استخدام الدوال وتنفيذها واستخدام متغيرات في كامل المشروع للتحكم بها .. إلخ أمور جداً كثيرة وتحتاجها لكي تزداد خبرتك ببرمجة تطبيقات أكثر من متوسطة.

مع إنشاء المشاريع سوف أركز أكثر على الأكواد إن شاء الله .

دوال دورة حياة الاكتفتي

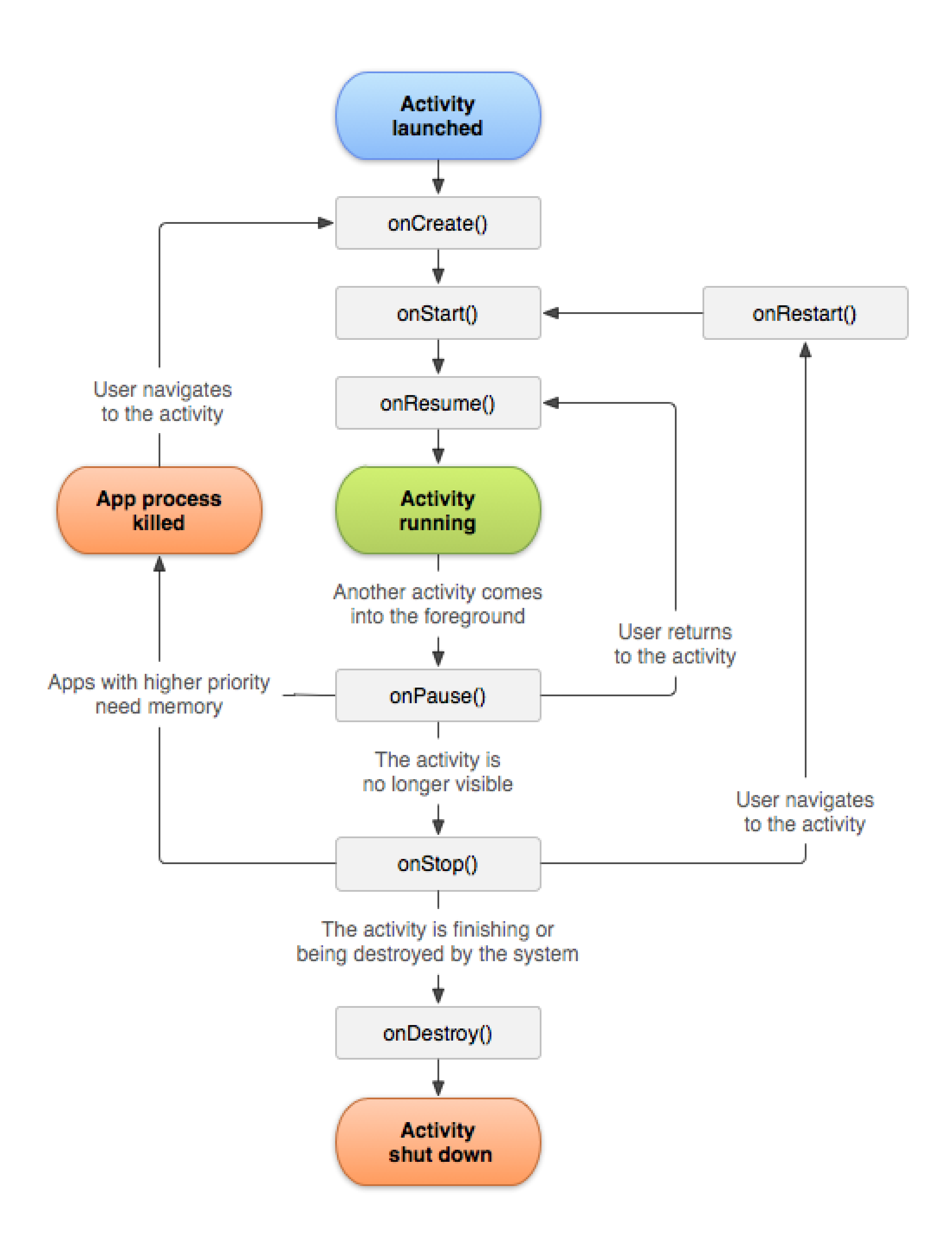
هنا سوف نتعلم الدوال الأساسية, التي سوف تنفذ دون استدعاء منك, وكل منها سوف ينفذ في موقع أو نقطة معينة, وتحتاجها وتستفيد منها لأنك لا تستطيع الوصول لهذه النقاط بسهولة. لذلك الأمر سهل عليك بعدة دوال أساسية للاكتفتي.

هل تتذكر في الاكتفتي شاهدنا داخلها دالة أساسية يتم إنشائها مع الاكتفتي وليس الكلاس ..

اعتقد هكذا وضح لك الأمر .. عند إنشاء كلاس لا تكون داخله دالة الاون كريت .. ولكن نضعها في الكلاس لتصبح اكتفتي يتم ربطه باللياوت ولتنفيذ عدة دوال معينة بالأندرويد ..

في البداية شاهد الصورة في الصفحة القادمة

هذه الصورة تعبر عن الدوال جميعها الأساسية للاكتفتي وأين موقع استخدمها :



إذا لم تفهمها جيداً تابع الشرح : http://goo.gl/KXeTp4

وهذه نبذة مختصر عن هذه الدوال :

الدالة	الوصف
onCreate():	الدالة الاساسية التي يتم استدعائها بعد انطلاق او تشغيل الاكتفتي, وهي الدالة التي يتم داخلها تعريف المتغيرات واعطاء الاوامر البرمجة إلخ . ومن بعدها يتم تنفذ دالة اون ستارت
onRestart():	يتم تنفيذ الأوامر في هذه الدالة .. بعد إن يتوقف تطبيقك عن العمل عند تشغيل التطبيق مرة اخرى .. ومن بعده يتم تنفذ دالة اون ستارت
onStart():	ويتم تنفيذ هذه الدالة بعد إن يصبح التطبيق ظاهر أمام المستخدم ومن بعدها يتم تنفيذ دالة اون ريزوم إذا كنت ما زلت داخل التطبيق, أو يتم تنفيذ دالة اون ستوب إذا ذهبت أو فتحت تطبيق آخر
onResume():	يتم تنفيذها عندما يبدأ المستخدم بالتفاعل مع التطبيق, ويتم تنفيذ من بعدها اون باز
onPause ():	تحدث هذه الدالة عندما يتم الرجوع للتطبيق إذا كنت في نشاط اخر وتأكد إنك لا تضع أوامر كثيرة هنا
onStop():	تحدث هذه الدالة عندما يصبح التطبيق غير ظاهر للمستخدم
onDestroy():	وتحدث هذه الدالة عندما يتم إجبار التطبيق أو توقفه بشكل اجباري

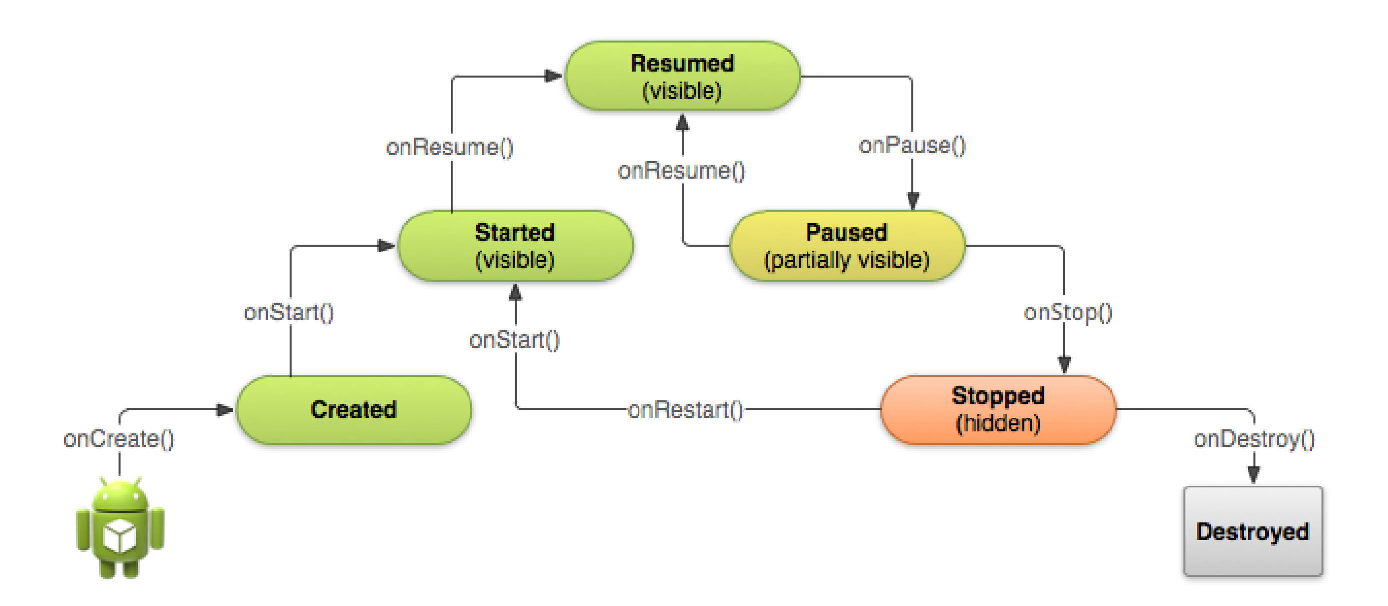
مثال صغير :

مثلاً يوجد قواعد بيانات في التطبيق لتخزين كم كبير من الرسائل, ويتم إنشاء القاعدة مع بداية تشغيل التطبيق, إذا يفضل عندما يتم الخروج من التطبيق إي تستخدم دالة التوقف والتوقف الاجباري لتفريغ هذه القاعدة, كي لا يصبح الحمل ثقيل على الجهاز, وأحياناً يتم ملأ الرسائل مع كل عملية فتح على حسب خبرتك في التعامل مع القواعد.

طريقة كتابة أحد الدوال السابقة :

```
public void onDestroy() {
        super.onDestroy();
// هنا يتم تنفيذ الأوامر .
    }
```

وهذه الصورة توضح لكم الدوال بشكل آخر :



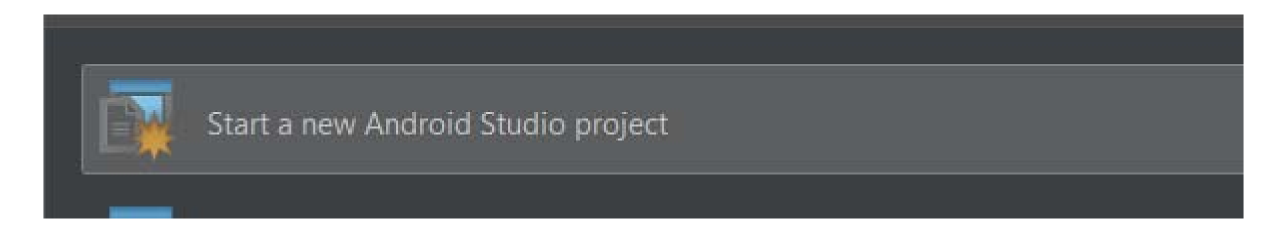
المشروع الأول 1- مرحباً يا ..

الآن سوف نقوم بالخطوة, الكثير منكم ينتظرها وهي بناء مشروع بسيط, سوف يكون عبارة عن حقل نصي وزر , تقوم بادخال اسمك ومن ثم الضغط على الزر, يتم الانتقال للاكتفتي الاخرى وعرض اسمك الذي قمت بإدخاله.

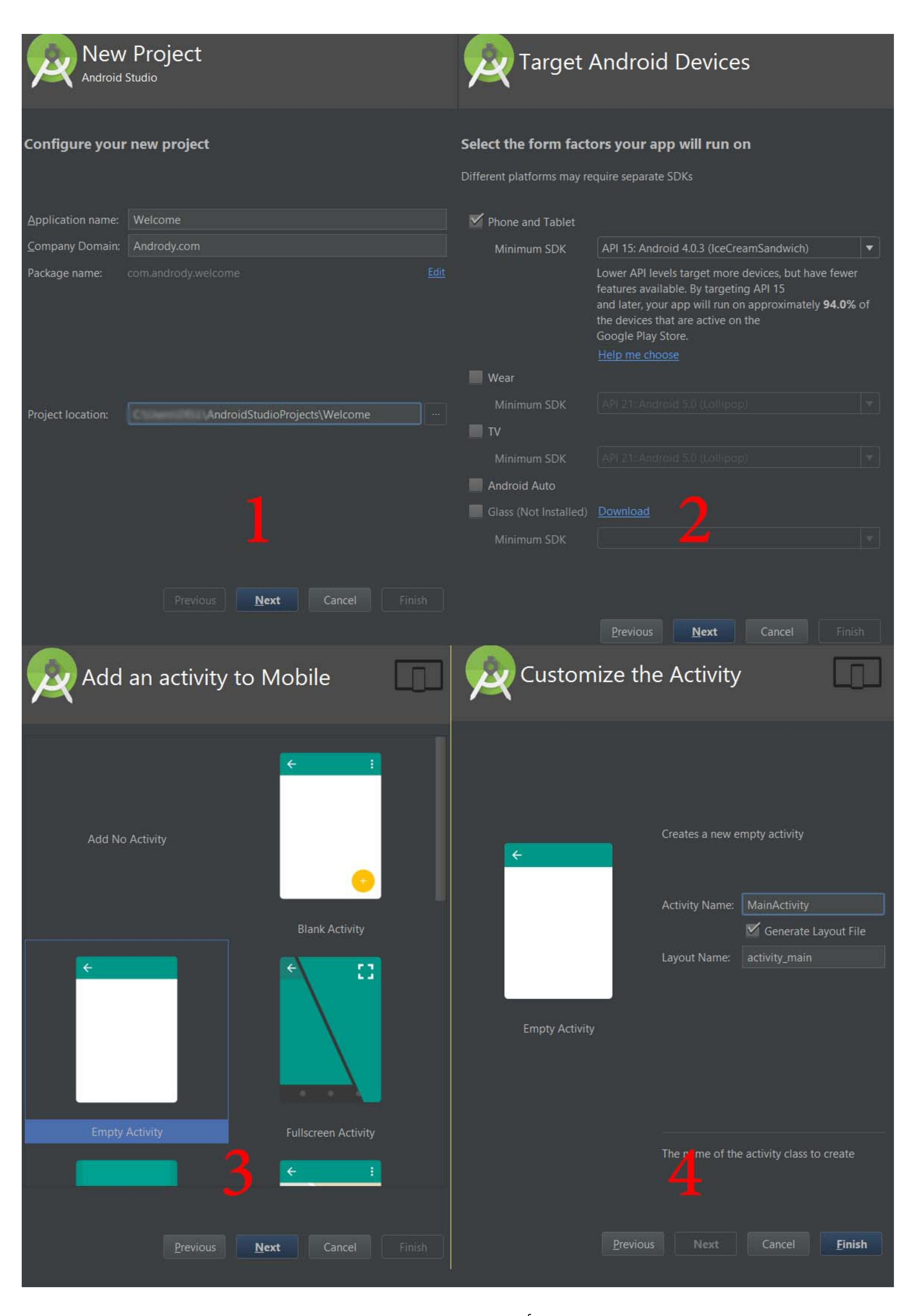
ومن هذا المشروع سوف نتعلم عدة أمور, منها وضع عناصر على الشاشة, تنفيذ أوامر بعد الضغط على الزر, إنشاء اكتفتي اخرى ولياوت اخرى, التعامل مع الحقل النصي وجلب ما بداخله, كيفية الانتقال من اكتفتي الى اخرى, وامور اخرى ..

وشوف نأخذ الخطوات بالترتيب وبالصور .. لأنه تقريباً للبعض سوف يكون أول مشروع له.

ولكن بما إننا أخذنا في درس سابق إنشاء اول مشروع لك, سوف أضع صور فقط سريعة من ثم نبدأ الشرح.



في البداية نقوم بفتح الأندرويد ستوديو .. والضغط على إنشاء مشروع جديد .. تابع الصور.

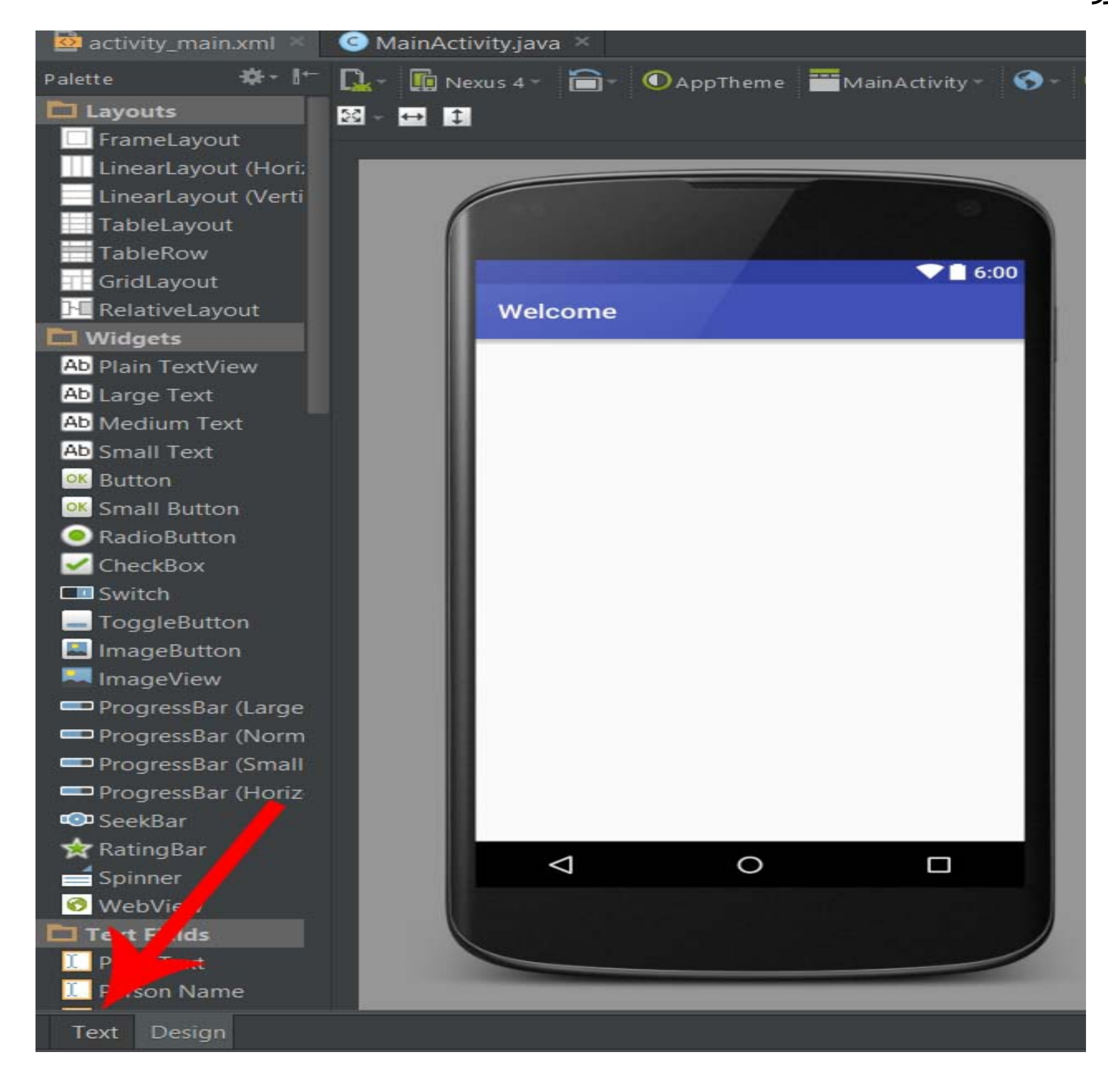


والآن بعد الضغط على زر Finish سوف يبدأ المشروع بالتجهيز ..

بعد فتح المشروع سوف نشاهد إن هناك ملفين مفتوحين .. الاكتفتي والليـاوت.

نبدأ مع الليـاوت نقوم بحذف النص الموجود Hello World! , وذلك بالنقر عليه ثم الضغط من لوحة المفاتيح على زر Delete.

بعد ذلك من نقوم بالضغط على Text الموجودة في الزاوية اليسرى في الأسفل من الليـاوت. كما في الصورة :



لكي يتم فتح لنا هذه النافذة ولكن بالعرض النصي او شاشة التكسـت فيو او باللغة البرمجية XML,

سوف يظهر لنا أكواد معدودة نقوم بحذف الأسطر التالية منها :

```
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
android:paddingBottom="@dimen/activity_vertical_margin"
```

هذه الأسطر هي خاصية الحشو , وذلك لجعل هامش ما بين اطراف الشاشة والعناصر داخلها, ولكن لا نحتاجها حالياً.

ونقوم كذلك بتغيير الكلمة التالية : RelativeLayout إلى LinearLayout.

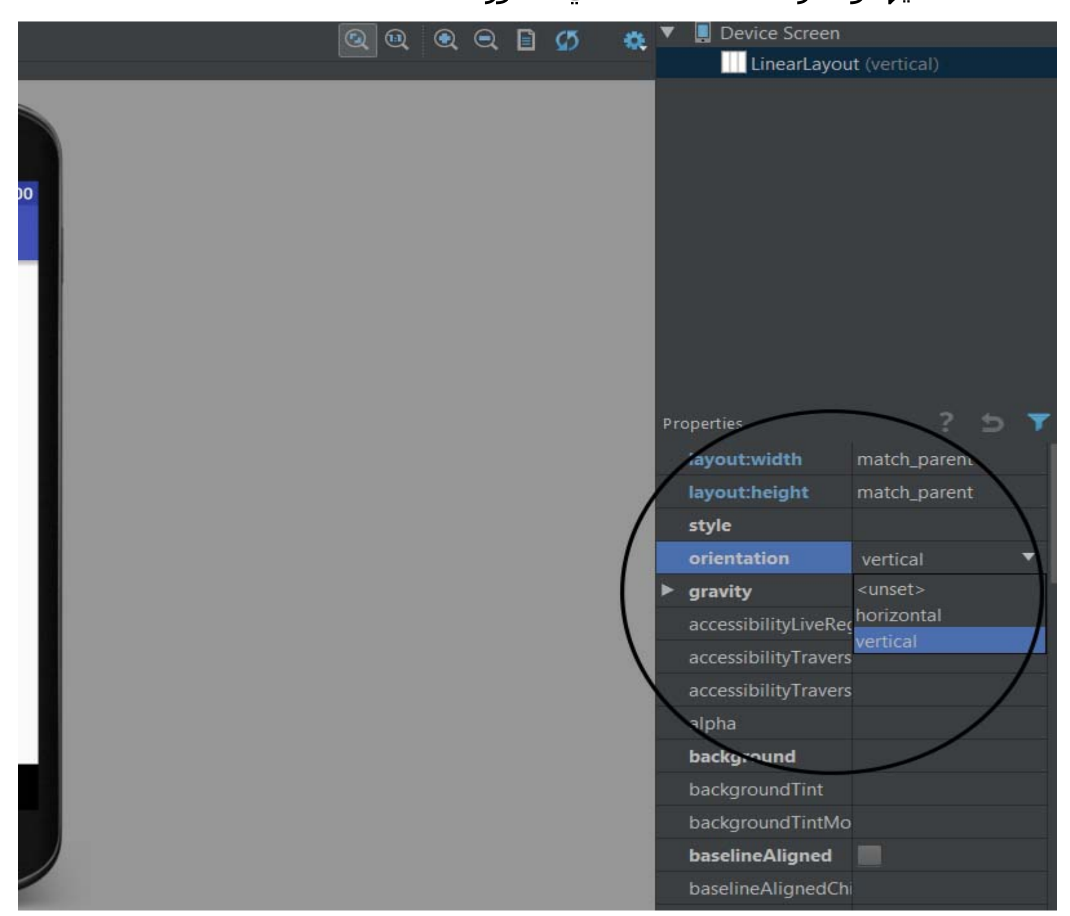
ليصبح الكود بالشكل التالي :

```
<?xml version="1.0" encoding="utf-8"?>
    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

</LinearLayout>
```

الآن نرجع إلى شاشة العرض .. الديزاين فيو , كذلك من الزاوية اليسرى في الأسفل انقر على Desgin .

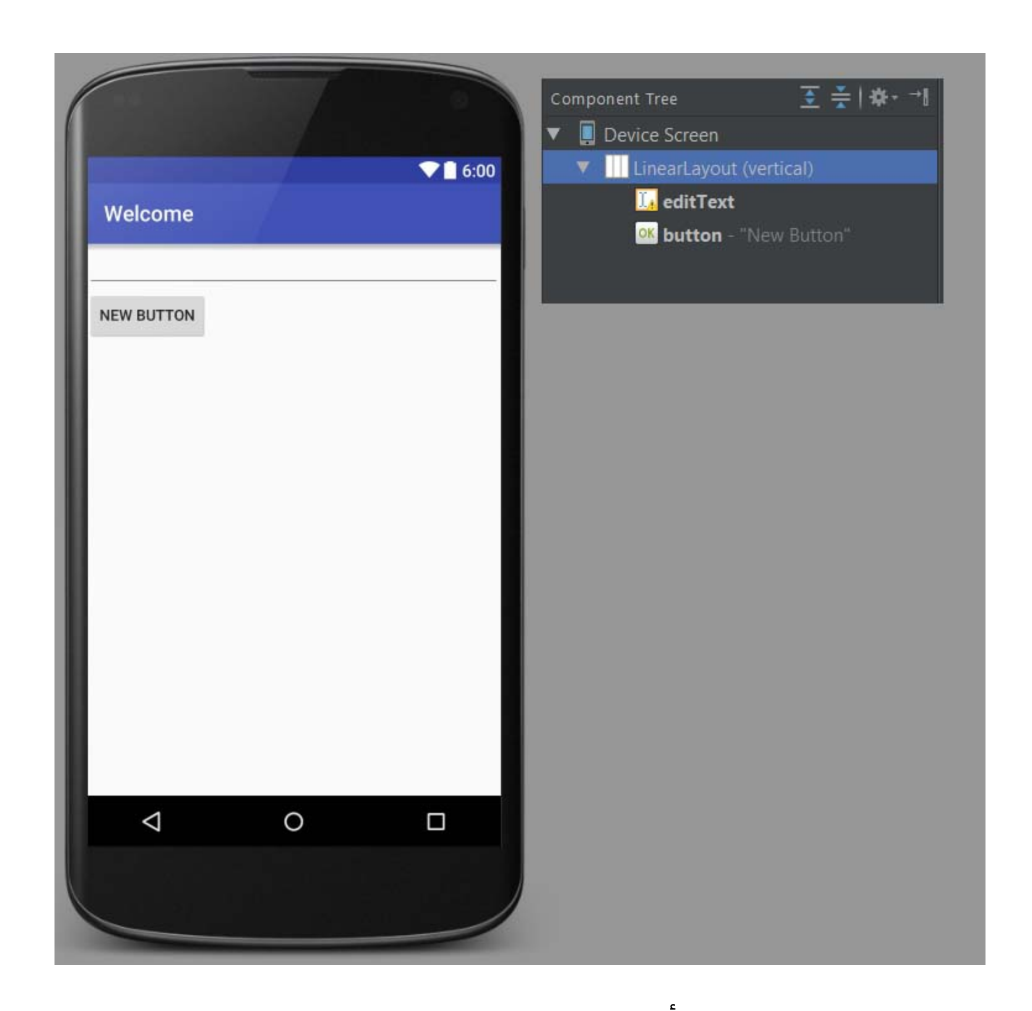
بعد الرجوع إلى شاشة الديزاين فيو, يوجد في الأسفل على اليمين الخصائص, يوجد خاصية اسمها orientation اضغط عليها واختر vertical. كما في الصورة :



هكذا قمنا بتحديد اتجاه اللينر لياوت بالأتجاه العمودي, لكي تصبح العناصر فوق بعضها البعض.

والآن سوف نقوم بالضغط على عنصر Plain Text وهو من نوع EditText بعد الضغط عليه فقط قم بالنقر على اللياوت, ثم نقوم بالضغط على عنصر Button وكذلك نقوم بالنقر على اللياوت ليتم ادراجهم.

سوف يخرج لدينا الشكل التالي :



الآن سوف نقوم بتنسيق كل منها. نبدأ بالتحديد على عنصر اللياوت كما في الصورة السابقة, ومن قائمة الخصائص على اليمين, نذهب إلى خاصية padding ثم all ونضع داخلها 20dp.

ثم نقوم بالنقر على عنصر editText, ومن الخصائص نذهب غلى خاصية id ونضع داخلها put_name ثم خاصية hint ونضع داخلها "اكتب اسمك هنا من فضلك".

والآن نقوم بتحديد او الضغط على الزر, ومن الخصائص نقوم بتغير الـ id إلى enter. وكذلك عند خاصية layout:gravity ثم داخلها عند center نختار horizontal. وعند خاصية text نكتب تفضل بالدخول .

والنتيجة هي :



ونتيجة الكود xml :

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
  android:padding="20dp"
  tools:context=".MainActivity">
  <EditText
    android:id="@+id/put_name"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="اكتب اسمك هنا من فضلك" />
  <Button
    android:id="@+id/enter"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:text="تفضل بالدخول" />
</LinearLayout>
```

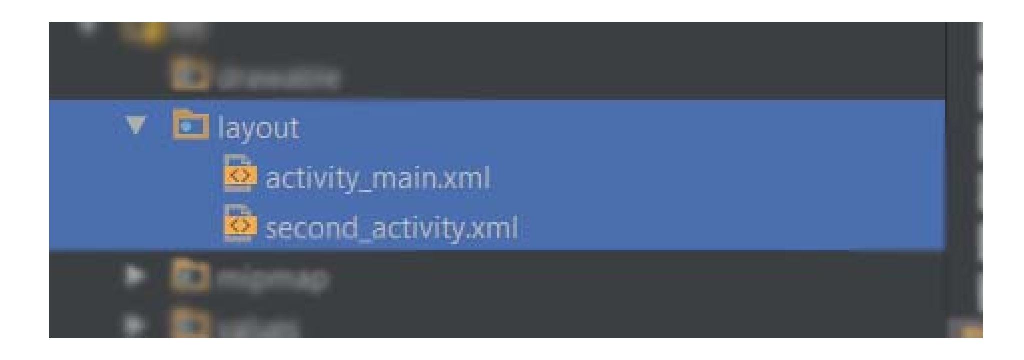
والآن لنكمل اللياوت أفضل, من ثم ننتقل للجانب البرمجي. ما سوف نقوم بفعله الآن, هو إنشاء لياوت آخرى, ليتم الانتقال إليها بعد الضغط على زر تفضل بالدخول.

لنقوم بإنشاء لياوت أخرى, نضغط بالزر الأيمن على مجلد layout ومن ثم new ثم Layout resource file



الآن سوف تظهر لك نافذة صغيرة, اترك الخيارات كما هي فقط حقل File name ضع فيه اسم اللياوت وسوف نضع نحن في هذا التطبيق : second_activity .

لدينا الآن لياوت اثنين :



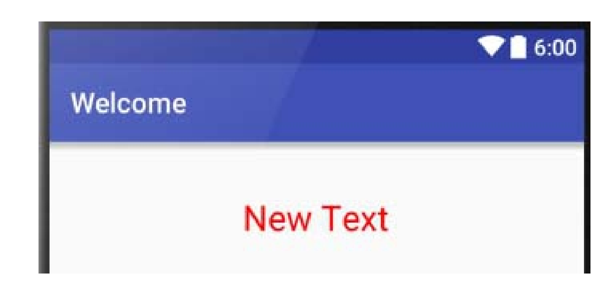
اللياوت الاولى قمنا بتجهيزها بالكامل, الآن سوف نقوم بفتح اللياوت الثانية.

سوف نقوم بوضع عنصر Plain TextView من نوع TextView, ونضغط عليه لنقوم بتعديل الخصائص الخاصة به.

من قائمة الخصائص نقوم في البداية تعديل الـ id إلى welcome . ثم خاصية layout:margin ثم من داخلها all ونضع القيمة 40dp . كذلك الخاصية layout:gravity ثم داخلها عند center نختار horizontal.

و الخاصية textColor نضع #f00 وعند الخاصية textSize نضع 25sp .

النتيجة هي :



والكود xml للياوت الثانية :

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
android:layout_width="match_parent"
  android:layout_height="match_parent">
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="New Text"
    android:id="@+id/welcome"
    android:layout_gravity="center_horizontal"
    android:layout_margin="40dp"
    android:textColor="#f00"
    android:textSize="25sp" />
</LinearLayout>
```

هكذا نكون أنتهينا من اللياوت .. سوف نذهب للجزء البرمجي بلغة الجافا java.

سوف نقوم أولاً بإنشاء الاكتفتي الثانية, من مجلد java حدد على الحزمة واضغط عليها بالزر الأيمن

واختر new ثم Java Class.



سوف تخرج نافذة صغيرة نضع الأسم التالي : Second_Activity

تماماً مثل اسم اللياوت ولكن اول حرف كبير, لا مشكلة بالتسمية ولكن للترتيب.

سوف يظهر لنا الكلاس بهذا الشكل :

```
package com.andrody.welcome;
/**
 * Created by Abboudi_Aliwi on 11/9/2015.
 */
public class Second_Activity {
}
```

فارغ تماماً, الآن يجب وضع داخله دالة الأون كريت ليصبح اكتفتي ويتم ربطه باللياوت الثانية.

من حفظ كيفية تعديل الكلاس وجعله اكتفتي ويضع داخله دالة الاون كريت هذا جيد, ولمن لا يعرفها يذهب إلى الاكتفتي الرئيسية واسمه MainActivity .

ومن داخله نقوم بنسخ الكود بالشكل التالي :

```
package com.andrody.welcome;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

و نقوم بلصق الكود في الكلاس الثاني, كما في الصورة :

```
MainActivity.java    Second_Activity.java
package com.andrody.welcome;

/**
 * Created by Abboudi_Aliwi on 11/9/2015.
 */
public class Second_Activity {
}
```

```
Cut                      Ctrl+X
Copy                     Ctrl+C
Copy as Plain Text
Copy Reference           Ctrl+Alt+Shift+C
Paste                    Ctrl+V
Paste from History...    Ctrl+Shift+V
Paste Simple             Ctrl+Alt+Shift+V
Column Selection Mode    Alt+Shift+Insert
```

بعد اللصق ...

```
MainActivity.java    Second_Activity.java
package com.andrody.welcome;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;

/**
 * Created by Abboudi_Aliwi on 11/9/2015.
 */
public class Second_Activity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

الآن بعد اللصق, لاحظ إن دالة الاون كريت مرتبطة باللياوت الرئيسية, لذلك سوف نقوم بتغيير السطر رقم 14 من :

```
setContentView(R.layout.activity_main);
```

إلى الليوات الثانية :

```
setContentView(R.layout.second_activity);
```

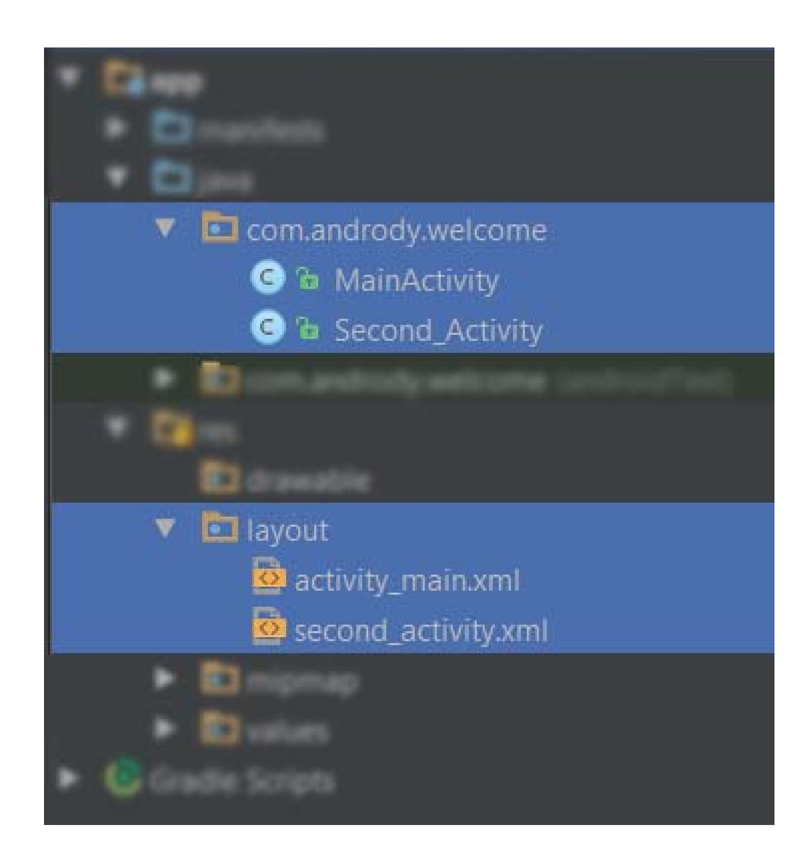
تمام, قبل أن نقوم بإضافة الأوامر البرمجية, نراجع ما لدينا الآن :

لدينا 2 اكتفتي + 2 لياوت.

وتم ربطها ببعضها.

كذلك تم الانتهاء من اللياوت.

باقي لدينا ملفات الاكتفتي لبرمجتها.



حسناً سوف نبدأ الآن مع الاكتفتي الاول : MainActivity

من داخله في البداية ما سوف نقوم به هو , إخبار الاكتفتي إن لدينا حقل نصي + زر, وربطها بالعناصر من داخل اللياوت عن طريق id وكذلك إعطاء أمر للزر قم بالانتقال بنا إلى الاكتفتي الثانية وخذ معك النص الذي وضعه المستخدم بداخل الحقل النصي.

كما تعلمنا سابقاً كيف نقوم بوضع المتغيرات, سوف نضع متغير من نوع زر ومتغير من نوع حقل نصي , نضعها خارج الدوال وبداخل الكلاس. نضع التالي :

```
EditText your_name;
Button go_to;
```

والآن بداخل دالة الاون كريت سوف نقوم بربط هذه المتغيرات بالعناصر داخل اللياوت بالشكل التالي :

```
your_name = (EditText)findViewById(R.id.put_name);
go_to = (Button)findViewById(R.id.enter);
```

و أسفل عملية الربط, نقوم بإنشاء دالة اون كليك للزر .. في حال تم الضغط عليه ليقوم بتنفيذ أوامر معينة. هذا الكود لبرمجة الزر :

```
go_to.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
          هنا نضع الأوامر البرمجية التي نريد القيام بها بعد الضغط على الزر //
        }
    });
```

و الآن نقوم بوضع الأوامر التالية :

```
Intent i = new Intent(MainActivity.this,Second_Activity.class);
        i.putExtra("text", your_name.getText().toString());
        startActivity(i);
```

قمنا بإعطاء أمر انتنت Intent وبالأندرويد بمعنى انتقال, سوف تتعرف على الأنتنت بشكل أكبر في درس مستقل.

هذا الأنتنت متغير اسمه i تستطيع كتابة ما تريد, ولكن حسب المثال كتبنا i ثم عرفنا إنه انتنت جديد وداخله عملية أنتقال من المكان أو الاكتفتي التي نحن فيها MainActivity.this إلى الاكتفتي الثانية Second_Activity.class . و أخبرنا الانتنت i إن يأخذ معه قيمة, وهي قيمة النص الموجود بداخل your_name وتحويلها إلى سترنج, وبعدها أعطينا أمر بداخل الزر لتنفيذ هذا الأنتنت.

هكذا نكون أنتهينا من الآن الاكتفتي الرئيسية, والآن ننتقل إلى الاكتفتي الثانية, ثم من بعدها سوف أطرح لكم أكواد الاكتفتي الاولى والثانية كاملة بعد الانتهاء.

بالنسبة لـ الاكتفتي الثانية : Second_Activity .

ما سوف نقوم به فقط, تغيير النص بداخله, ويجعله يستقبل النص من الاكتفتي الاولى ثم وضعه في هذا التكست فيو.

نقوم بفتح الاكتفتي الثانية, وبما إنه ليس لدينا أوامر برمجية ودوال كثيرة, لا مشكلة في دالة الاون كريت نقوم بتعريف النص وربطه ومن ثم اعطائه أمر ليستقبل النص الجديد.

في دالة الاون كريت ضع التالي :

TextView **Welcome** = (TextView) findViewById(R.id.welcome);

للمراجعة Welcome هو المتغير من نوع TextView , والعنصر المعرف id الخاص به هو welcome .

و الآن سوف نقوم بوضع set للنص لتعديله كما ذكرنا سابقاً بالكود التالي :

Welcome.setText("مرحباً "+getIntent().getStringExtra("text"));

استخدمنا setText للنص بإسمه البرمجي الجديد Welcome وجعلنا النص الجديد داخله, Welcome + النص الذي جلبه الانتنت معه من الاكتفتي الاولى.

لم أتعمق بشرح ما سبق لأنها سهلة وسوف تتكرر معك كثيراً ومن خلال الاستخدام سوف تستخدمها.

كود الأكتيفتي الرئيسية : MainActivity

```
package com.andrody.welcome;

import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

  EditText your_name;
  Button go_to;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    your_name = (EditText)findViewById(R.id.put_name);
    go_to = (Button)findViewById(R.id.enter);
    go_to.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        Intent i = new Intent(MainActivity.this,Second_Activity.class);
        i.putExtra("text", your_name.getText().toString());
        startActivity(i);
      }
    });
  }
}
```

وأما كود الاكتفتي الثانية : Second_Activity

```java
package com.andrody.welcome;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.widget.TextView;

/**
 * Created by Abboudi_Aliwi on 11/9/2015.
 */

public class Second_Activity extends AppCompatActivity {

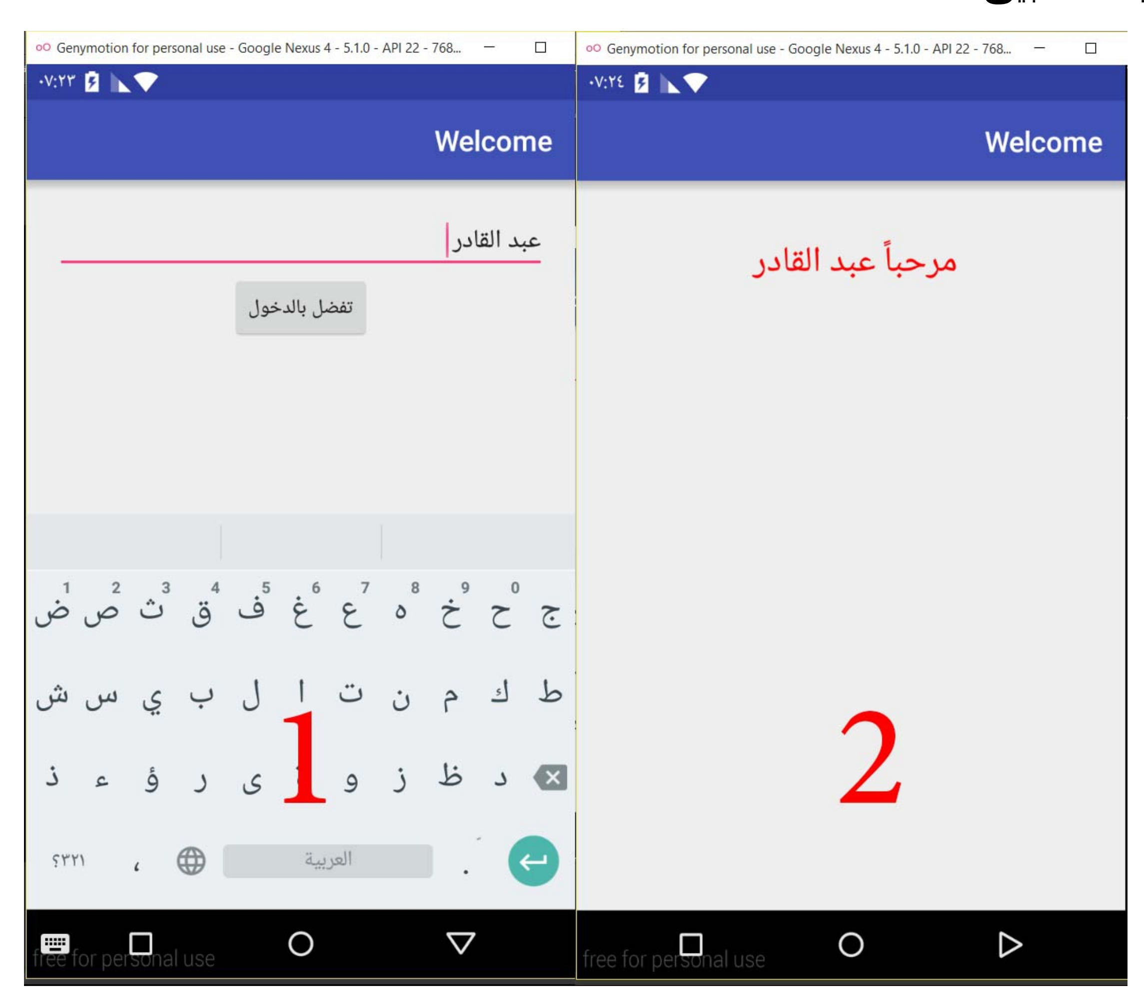
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.second_activity);

    TextView Welcome = (TextView) findViewById(R.id.welcome);

    Welcome.setText(" مرحبًا "+getIntent().getStringExtra("text"));

  }
}
```

و نتيجة التطبيق :



تنسيق تطبيق الأندرويد

في هذا الدرس إن شاء الله, سوف نأخذ طرق بسيطة وصغيرة, لإعطاء لمسة أكثر جمالية للتطبيق.

النص طويل :

إذا كان لديك نصوص كثيرة, وتريد إضافتها في تطبيقك, سوف أساعدك هنا بعدة خيارات, اختر ما يناسبك منها, إذا كان النص جداً مهم. ضع كل فقرة أو سطر منه, داخل TextView او برمجياً من الجافا String لكل سطر, في حال شخص أردات سرقة المحتوى سوف يجد صعب في فك تشفير التطبيق وإن استطاع سوف يجد كل سطر داخل TextView ولكل منها خصائص مختلفة لذلك سوف يجد ملل شديد في جمع النصوص الكثيرة, إما لتنسيق النص, إذا كانت كل واحد منها منفصلة سوف تجد صعوبة في التنسيق, لانك تحتاج بالتعديل على واحد منها, لذلك لسهولة وضع النصوص الطويلة أستخدم بعد كل فقرة علامة (n\) في ملفات xml . لتجعل النصوص في فقرات وليست متتالية.

كذلك أغلب المبتدئين يعاني من عدم ظهور كامل النص, لذلك يجب عليك استخدام اداة او عنصر من نوع ScrollView ليتم عرض النص بأكلمه بخاصية التمرير للأسفل والاعلى.

نص يحتوي على روابط أو أرقام - ايميلات إلخ :

إذا كان لديك نص يحتوي على أرقام هواتف, أو روابط انترنت, أو إيميلات, أو احداثيات خرائط إلخ.

استخدم الخاصية التالية للنص :

```
android:autoLink="all"
```

ليسمح للمستخدم الضغط عليها.

اجبار المستخدم على تعبئة الحقل النصي :

إذا كان يوجد لديك حقل نصي او فراغ نصي EditText وتريد مثلاً من المستخدم ادخال اسمه, ولا يستطيع الضغط على المتابعة دون ملأ هذا الحقل, تستخدم شرط if , وتستطيع تحديد عدد الاحرف التي تريدها, أو الاسم او الكلمة التي تريد من المستخدم كتابتها إلخ.

ما يهمنا هنا إن لا يكون فاضي تماماً يجب كتابة إي شيء داخله, نستطيع شرط التحقق من الحقل

أن لا يكون فارغ. بالشكل التالي :

```
if(your_name.getText().toString().equals("")){
        // الاوامر التي يتم تنفيذها إذا كان الحقل فارغ
    }else {
        // الاوامر التي يتم تنفيذها إذا لم يكن الحقل فارغاً
    }
```

علماً أن **your_name** هو المتغير الخاص بالحقل النصي. ويوجد طرق اخرى كذلك.

إظهار رسالة توست او رسالة تنبيه مصغرة :

رسالة التوست هي التي تظهر على الشاشة بشكل مؤقت, وغالباً ما تكون أسفل الشاشة.

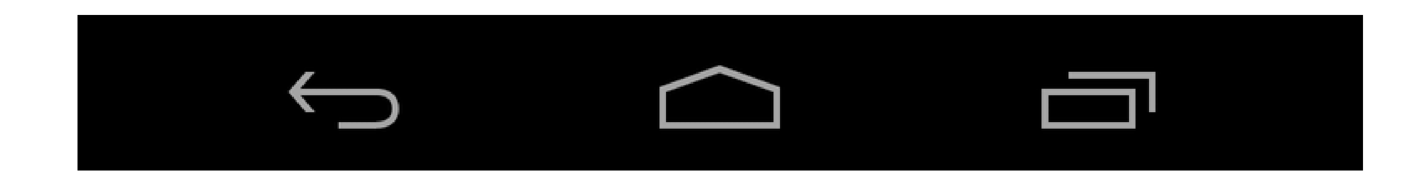
تستطيع إنشائها بشكل سريع فقط كتابة Toast داخل الأكواد البرمجية الجافا, في الأندرويد ستوديو وسوف يعطيك اقتراح Create a new Toast اضغط عليه وسوف يتم انشائها بشكل تلقائي.

وهي بالشكل التالي :

```
Toast.makeText(MainActivity.this, "اكتب الرسالة التي تريدها هنا", Toast.LENGTH_SHORT).show();
```

علماً إن MainActivity هو اسم الاكتفتي التي نحن فيها الآن, و LENGTH_SHORT هي المدة الزمنية التي يحتاجها التوست للإختفاء بمعنى مدة ظهور التوست و LENGTH_SHORT = من الثواني (2) وأما LENGTH_LONG = من الثواني (3.5) . وبطرق اخرى تستطيع إظهار الرسالة لمدة أطول إلخ.

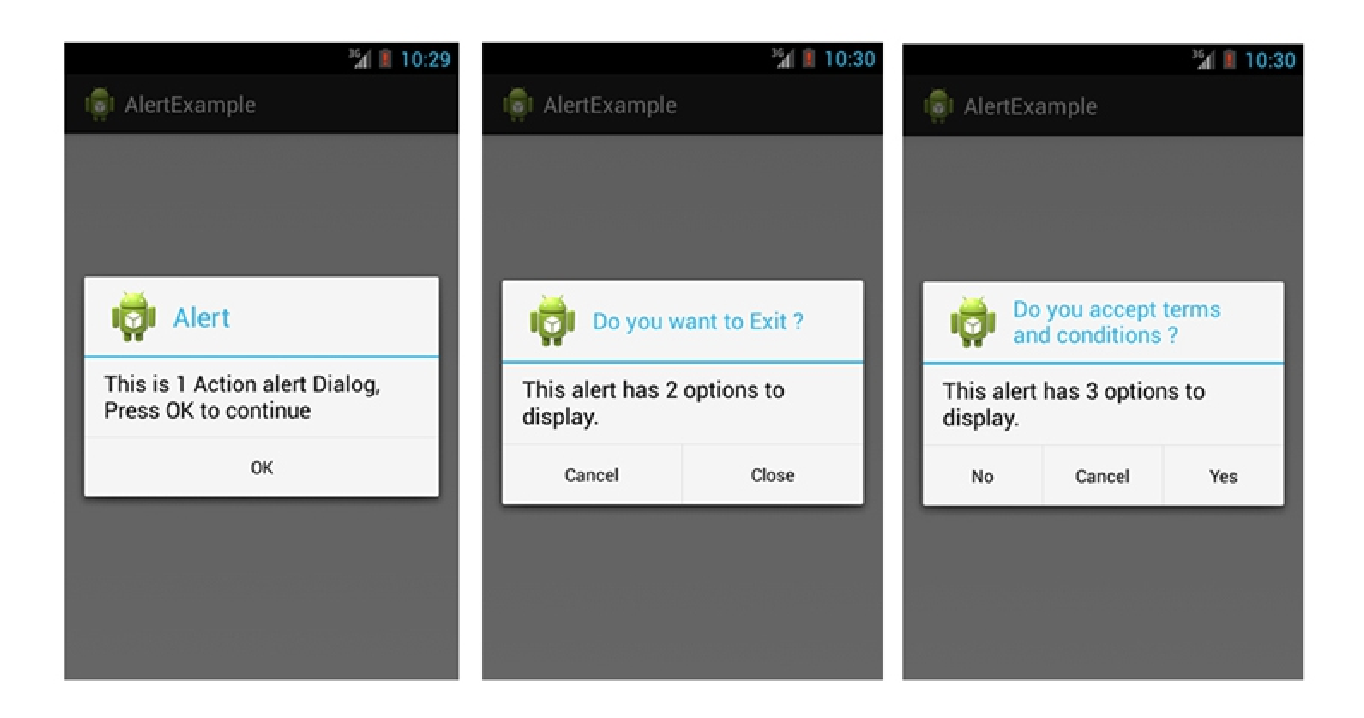
وصورة من هذه التوست لتتعرف عليها :



إظهار نافذة تنبيه ملاحظة alert dialog :

تطبيقات كثيرة تحتوي على هذه النافذة, من أمثلة استخدامها, هل متأكد من الخروج من التطبيق ؟ هل تريد تحميل .. ؟ هل متأكد من ؟ تحذير يوجد .. ! ملاحظة إلخ.

صورة للتعرف على هذه الناقذة :



هي عبارة عن أمر لظهورها, لذلك أنت مخير أنت تريد خروجها, بعد تشغيل اكتفتي, بعد الضغط على الزر إلخ.

الكود الخاص بها :

```
AlertDialog.Builder builder = new AlertDialog.Builder(
    MainActivity.this);
builder.setTitle("إغلاق التطبيق"); // العنوان
builder.setMessage("هل متأكد من خروج من التطبيق :"); // المحتوى
builder.setPositiveButton("نعم", // الزر الاول
    new DialogInterface.OnClickListener() {
      public void onClick(DialogInterface dialog,
              int which) {
        // الاوامر التي تحدث بعد الضغط على الزر الأاول
      }
    });
builder.setNegativeButton("لا", // الزر الثاني
    new DialogInterface.OnClickListener() {
      public void onClick(DialogInterface dialog,
              int which) {
        // اوامر الزر الثاني
      }
    });
builder.show();}
```

قم بتعديل الكود على راحتك, العنوان والمحتوى والازرار, وتستطيع تقليل او زيادة الازرار 1,3.

المشروع الأول 2- مرحباً يا .. متعدد اللغات

رح نكمل في هذا الدرس, تطوير مشروعنا الأول السابق, مرحباً يا .. وسوف نقوم ببعض الخطوات منها إجبار المستخدم على وضع اسم أقل شيء 3 احرف, بالإضافة إلى أن يكون التطبيق متعدد اللغات وسوف نجعله باللغة الإنجليزية ونضيف اللغة العربية, لأن اللغة الأساسية هي الإنجليزية.

في البداية لو رجعنا إلى الاكتفتي الرئيسية MainActivity , نجد إن هناك زر وداخله أوامر يتم تنفيذها بعد الضغط عليه, سوف نجعل هذه الأوامر بداخل شرط, أي يتم التحقق أولاً إن الحقل ليس فارغاً, ثم يتأكد أن الأحرف داخله أكثر من ثلاث ليتم تنفيذ هذا الأمر وهو الانتقال للاكتفتي الثانية.

برمجة الزر كانت :

```
go_to.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View v) {
    Intent i = new           Intent(MainActivity.this,Second_Activity.class);
    i.putExtra("text", your_name.getText().toString());
    startActivity(i);
  }
});
```

وسوف نجعلها هكذا :

```
go_to.setOnClickListener(new View.OnClickListener() {

  @Override
  public void onClick(View v) {

    if(your_name.getText().toString().isEmpty()){
      Toast.makeText(MainActivity.this, "من فضلك أكتب اسمك",
Toast.LENGTH_SHORT).show();

    }else if (your_name.getText().toString().trim().length() < 3){
      Toast.makeText(MainActivity.this, "عفواً يجب ادخال اسم صحيح",
Toast.LENGTH_SHORT).show();

    }else {
      Intent i = new Intent(MainActivity.this,Second_Activity.class);
      i.putExtra("text", your_name.getText().toString());
      startActivity(i);
    }
  }
});
```

لنراجع الكود السابق معاً :

في البداية إذا, (كان النص بداخل الحقل فارغا) قم بإظهار رسالة التوست من فضلك .. إلخ, وتوقف.

في حال لم يكن فارغاً إنتقل إلى الشرط الثاني, إذا (كان النص بداخل الحقل عدد الأحرف داخله أقل من 3 أحرف) إظهر رسالة التوست عفواً يجب إدخال .. إلخ. وتوقف.

إذا تخطى الشرطين السابقين, بمعنى إن الحقل ليس فارغاً وعدد الأحرف أكثر من 3 , ينفذ الأوامر التي طلبناها وهي الانتقال الى الاكتفتي الثانية.

```
if (your_name.getText().toString().isEmpty()){
        Toast.makeText(MainActivity.this, "من فضلك أكتب اسمك",
Toast.LENGTH_SHORT).show();
    }else if (your_name.getText().toString().trim().length() < 3){
        Toast.makeText(MainActivity.this, "عفواً يجب ادخال اسم صحيح",
Toast.LENGTH_SHORT).show();
    }else {
        Intent i = new   Intent(MainActivity.this,Second_Activity.class);
        i.putExtra("text", your_name.getText().toString());
        startActivity(i);
    }
```

تستطيع عكس الشرط بوضع علامة التعجب ! قبل الشرط, بمعنى عكسه. بمعنى الشرط إذا كان

الحقل فارغ, علامة التعجب قبل الشرط معناها إذا لم يكن الحقل فارغاً.

والآن رح ننتقل لكيفية جعل التطبيق أكثر من لغة, سوف نضيف اللغة العربية له, وتكون اللغة الإنجليزية الافتراضية لجميع الأجهزة ولكن لو كانت لغة الجهاز عربية, سوف تتغير النصوص للعربية.

في البداية رح نرجع للتطبيق وتغيير جميع النصوص للغة الإنجليزية. أما النصوص في الجافا فيجب إضافتها إلى ملف strings.xml .

نقوم بفتح هذا الملف. ما داخله :

```
<resources>
    <string name="app_name">Welcome</string>
</resources>
```

app_name / اسم التطبيق الذي يظهر على جهاز المستخدم.

سوف نقوم بإضافة جميع النصوص هنا بنفس الطريقة. وهي كالتالي :

```
<resources>
    <string name="app_name">Welcome App</string>
    <string name="text1">Write your name here please</string>
    <string name="text2">log in</string>
    <string name="text3">Please enter your name</string>
    <string name="text4">Please enter a valid name</string>
    <string name="text5">Welcome</string>
</resources>
```

المتغيرات أو الأسماء لك حرية اختيارها, وأما القيمة فهي النص الذي سوف يظهر.

بعد نهاية الدرس سوف أعرض لكم أكواد الدرس بالكامل, ومنها سوف تتعرف أين استخدامنا المتغيرات هذه text1,text2 إلخ.

بداخل ملفات xml بالصورة التالية :

```
@string/text1
```

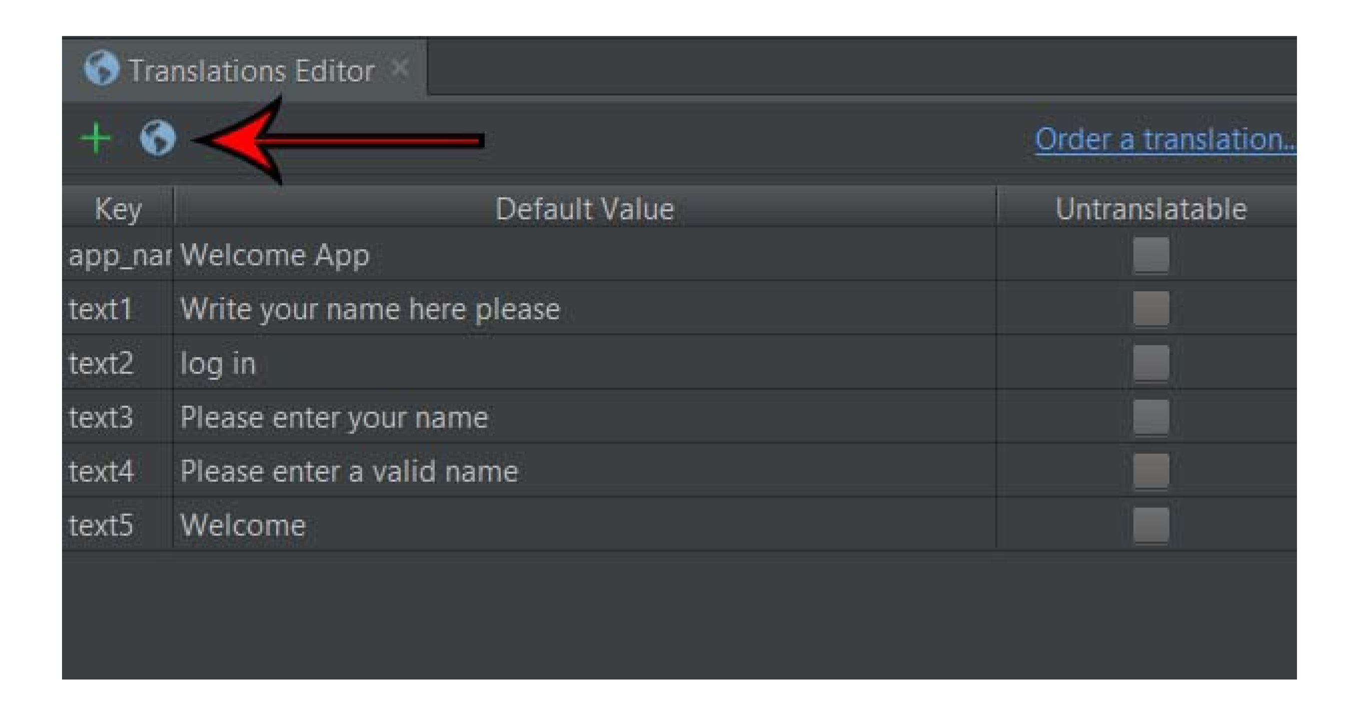
بداخل ملفات java بالصورة التالية :

```
getString(R.string.text3)
```

والآن سوف نقوم بإضافة اللغة العربية, نقوم بفتح ملف styles.xml , وفي الاعلى يوجد شريط تنبيه صغير, اضغط على open editor.

```
Edit translations for all locales in the translations editor.                    Open editor  Hide notification
<resources>
    <string name="app_name">Welcome App</string>
    <string name="text1">Write your name here please</string>
    <string name="text2">log in</string>
    <string name="text3">Please enter your name</string>
    <string name="text4">Please enter a valid name</string>
    <string name="text5">Welcome</string>
</resources>
```

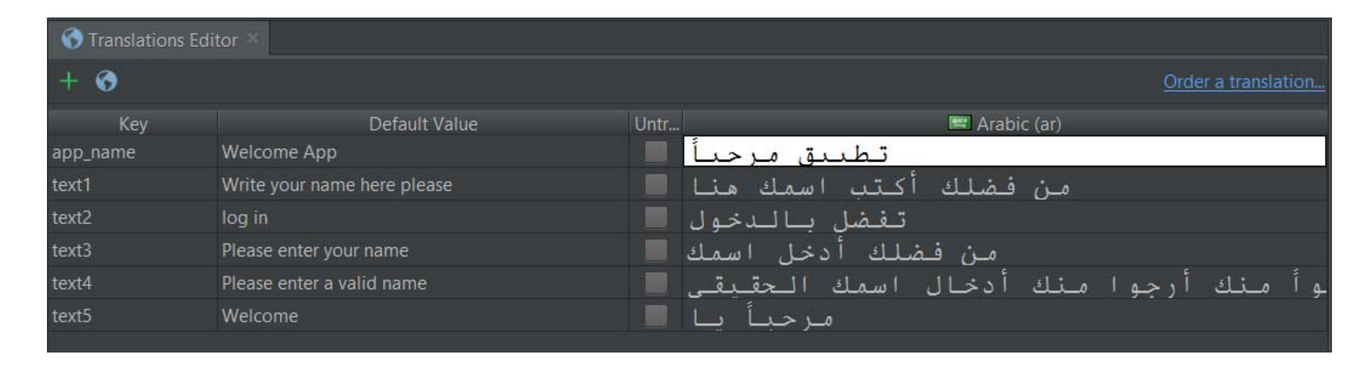
سوف تفتح لك نافذة :



يوجد داخلها في الاعلى يمين, صورة كرة ارضية صغيرة قم بالنقر عليها واختر اللغة العربية.



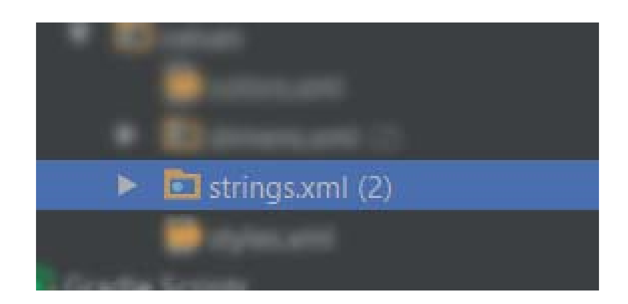
سوف يتم إضافة عمود جديد للغة العربية, قم بكتابة الكلمات من جديد باللغة العربية, بهذا الشكل :



Translations Editor

Key	Default Value	Untr...	Arabic (ar)
app_name	Welcome App		تطبيق مرحباً
text1	Write your name here please		من فضلك أكتب اسمك هنا
text2	log in		تفضل بالدخول
text3	Please enter your name		من فضلك أدخل اسمك
text4	Please enter a valid name		و أ منك أرجوا منك أدخال اسمك الحقيقي
text5	Welcome		مرحباً يا

فقط هكذا انتهينا .. سوف تلاحظ أن ملف styles.xml أصبح أثنين لديك في المشروع :

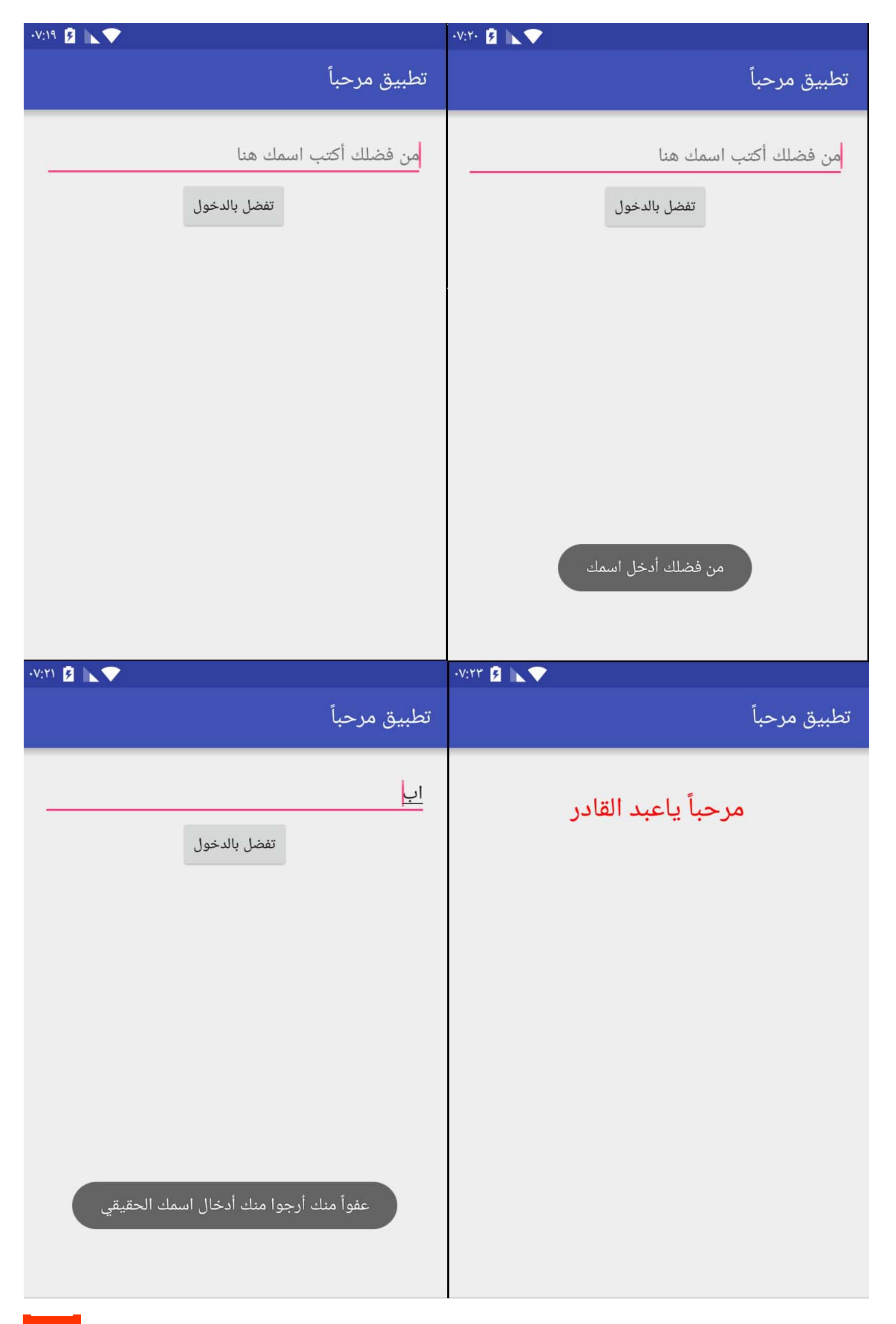


واحد منها للغة العربية والآخر للغة الإنجليزية .

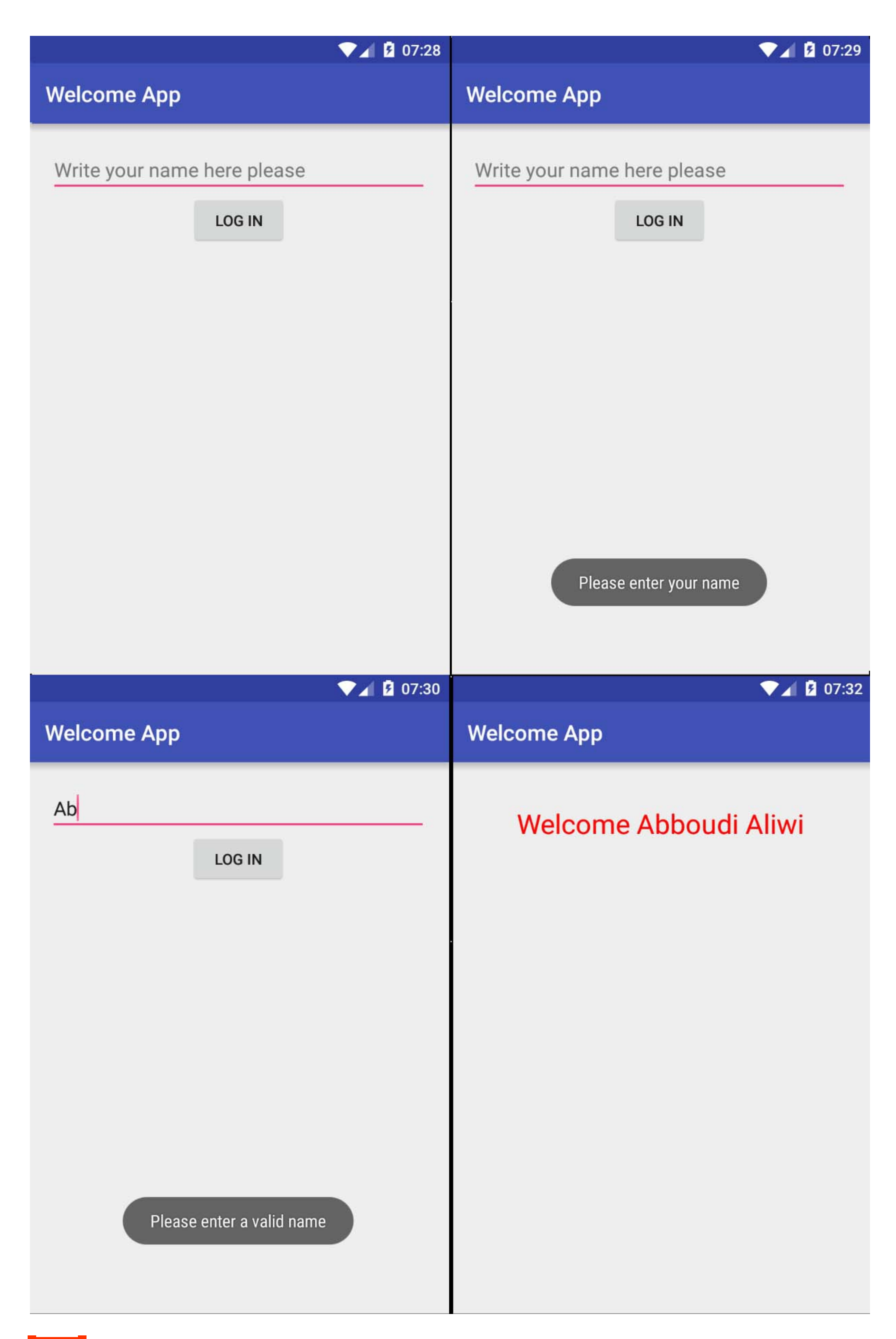
```
strings.xml   ar\strings.xml
Edit translations for all locales in the translations editor.
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">تطبيق مرحباً </string>
    <string name="text1">من فضلك أكتب اسمك هنا</string>
    <string name="text2">تفضل بالدخول</string>
    <string name="text3">من فضلك أدخل اسمك</string>
    <string name="text4">عفوا منك أرجوا منك أدخال اسمك الحقيقي</string>
    <string name="text5">مرحباً يا </string>
</resources>
```

```
values\strings.xml   ar\strings.xml
Edit translations for all locales in the translations editor.
<resources>
    <string name="app_name">Welcome App</string>
    <string name="text1">Write your name here please</string>
    <string name="text2">log in</string>
    <string name="text3">Please enter your name</string>
    <string name="text4">Please enter a valid name</string>
    <string name="text5">Welcome</string>
</resources>
```

والآن سوف أعرض النتيجة على جهاز لغته العربية وآخر لغته الإنجليزية :



٠٧:١٩
تطبيق مرحباً
من فضلك أكتب اسمك هنا
تفضل بالدخول
٠٧:٢٠
تطبيق مرحباً
من فضلك أكتب اسمك هنا
تفضل بالدخول
من فضلك أدخل اسمك
٠٧:٢١
تطبيق مرحباً
اب
تفضل بالدخول
عفواً منك أرجوا منك أدخال اسمك الحقيقي
٠٧:٢٢
تطبيق مرحباً
مرحباً ياعبد القادر


07:28
Welcome App
Write your name here please
LOG IN
07:29
Welcome App
Write your name here please
LOG IN
Please enter your name
07:30
Welcome App
Ab
LOG IN
Please enter a valid name
07:32
Welcome App
Welcome Abboudi Aliwi

بصراحة لو قمت بعرض الأكواد هنا, سوف يأخذ الكتاب صفحات زيادة, لذلك من الأفضل أن أضع لكم رابط لتحميل المشروع بشكل مباشر.

المشروع الأول : مرحباً يا .. متعدد اللغات.

من هذا المشروع نستفيد من :

- كيفية تعريف المتغيرات.

- كيفية برمجة الزر إذا تم الضغط عليه.

- جلب النص الذي بداخل الحقل النصي.

- وضع شرط if كذلك أكثر من شرط يتم التحقق منها.

- الانتقال من واجهة إلى أخرى.

- إضافة لغات للتطبيق.

- نقل بيانات من واجهة إلى أخرى.

- التحقق من الحقل إذا كان فارغاً أو لا.

- التحقق من عدد الأحرف.

لتحميل المشروع : http://goo.gl/RMgXGT

كيف أقوم ببناء مشاريع أندرويد بخطوات صحيحة

لتقوم في بناء برمجة تطبيقات الأندرويد, من الأفضل لك أن تتبع عدة طرق لكي تتجنب أغلب المشاكل والأخطاء أثناء بناء المشروع.

ومن أفضل الطرق هي استخدام القلم لتسجيل كل ما سوف تقوم به, وتعيين ما قمت به, وما الذي يتوجب عليك القيام به. والرسم البياني هو الأفصح و الأفضل لبناء المشاريع.

مثال نريد القيام ببناء تطبيق للأذكارات أو الأدعية, وهذا أكثر تطبيق أفضل أن يقوم بدراسته المبتدئين.

تم شرح بناء التطبيق في دروس فيديو مسجلة على اليوتيوب , لذلك أتمنى متابعتها للمبتدئين حصراً , سوف تستفيد منها بالتأكيد, ونظراً لاختصار الكتاب عدم ذكر الطريقة كاملة هنا.

1- الجزء الأول// تمهيد وتجهيز للبدء ببرمجة تطبيق اندرويد بسيط للمبتدئين, يحتوي على خصائص عديدة يستفيد منها كل مبتدئ في كيفية بناء تطبيقه الأول كاملاً.

https://youtu.be/RqOQTvREJsw

2- الجزء الثاني// مخطط ورسم للتطبيق وتجهيزيه قبل البدأ ببرمجته والعمل عليه. مهم جداً ~

https://youtu.be/DHI1z0MH5Qw

3- الجزء الثالث// إنشاء الاكتفتي واللياوت المطلوبة وربطها ببعضها.

https://youtu.be/U_bqbLN3Exo

4- الجزء الرابع// تجهيز الرئيسية ووضع الازرار عليها وبرمجتها.

https://youtu.be/M-gqL6BjD4s

5- الجزء الخامس// تجهيز وتنسيق الصفحة الاولى الخاصة بالاذكارات.

https://youtu.be/h_7mhy6DeJI

6- الجزء السادس// برمجة صفحة الاذكارات, تشغيل وايقاف الصوت, تكبير وتصغير النص.

http://youtu.be/SJR8lpZkyg0

7- الجزء السابع// تجهيز وبرمجة صفحة حول التطبيق, لينك الموقع, رقم الهاتف, الايميل, تقييم التطبيق, المزيد من تطبيقات المبرمج.

https://youtu.be/P0O_0a2nx2E

8- الجزء الثامن// الجزء الأخير في برمجة تطبيق الاذكارات, تحميل المشروع كامل مفتوح المصدر, كود تأكيد اغلاق التطبيق, تصدير المشروع لملف apk جاهز للرفع للمتجر.

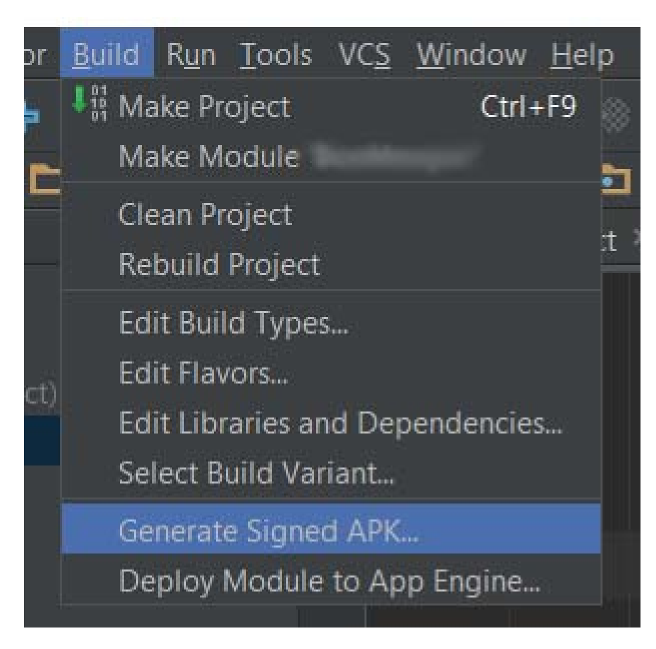
https://youtu.be/ld6f87xqJUc

تصدير مشروعك إلى تطبيق أندرويد Apk

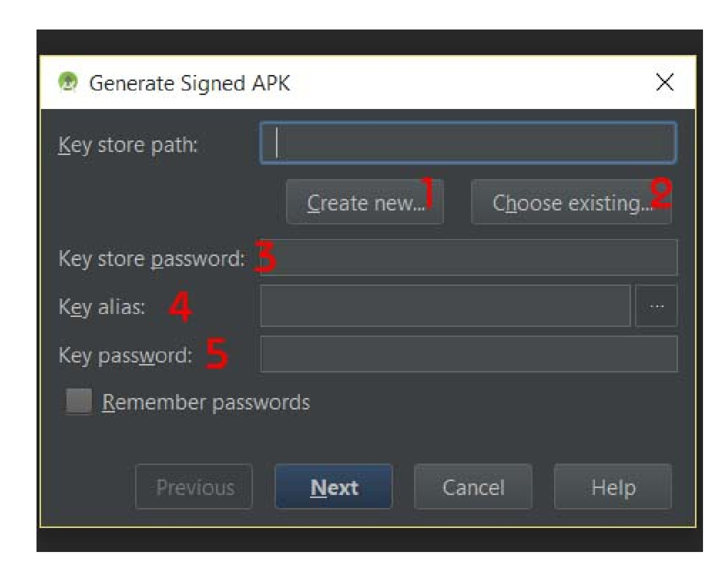
بعد الانتهاء من المشروع وتريد ترسل ملف apk لأحد الأصدقاء لتجربته, او ارساله إلى جهازك, توجد إلى مجلد المشروع ومسار المشروع تجده أعلى النافذة, او من خصائص المشروع او المسار الذي تم إنشاء التطبيق عليه في بداية انشاء التطبيق, توجد إلى مسار المشروع يوجد مجلد إسمه app بداخله توجه للمسار \build\outputs\apk

سوف تجد ملف apk بعنوان - اسم المشروع + debug.apk . قم بإرساله إلى إي جهاز وقم بتجربته, ولكن لن تستطيع رفعه للمتجر.

لكي تقوم بتصدير المشروع ورفعه كتطبيق على المتجر, بعد الإنشاء من المشروع, من أعلى البرنامج اختر نافذة Build ثم Generate Signed APK



الآن سوف تظهر لك نافذة :



في هذه النافذة سوف نقوم بإنشاء Key Store مرة واحدة للمبرمج, و Key alias لكل تطبيق خاص بنا.

على راحتك تستطيع إنشاء أكثر من Key Store. عموماً لكن هي الفكرة تعتبر نوعاً ما حماية لتطبيقك, مثلاً Key Store وهي ملف بصيغة jsk. بعد القيام بإنشائه, تقوم بحفظ مفتاح لكل تطبيق داخل هذا الملف, في حال قمت بتعديلات إجباري تقوم بحفظ التعديل على نفس المفتاح. لو شخص ما تسلسل إلى حسابك لن يستطيع إعادة رفع تطبيق على تطبيقك بهذه الطريقة.

عموماً لنكمل, حسب الأرقام الموضحة في الصورة السابقة :

1/ لإنشاء شفرة Key Store.

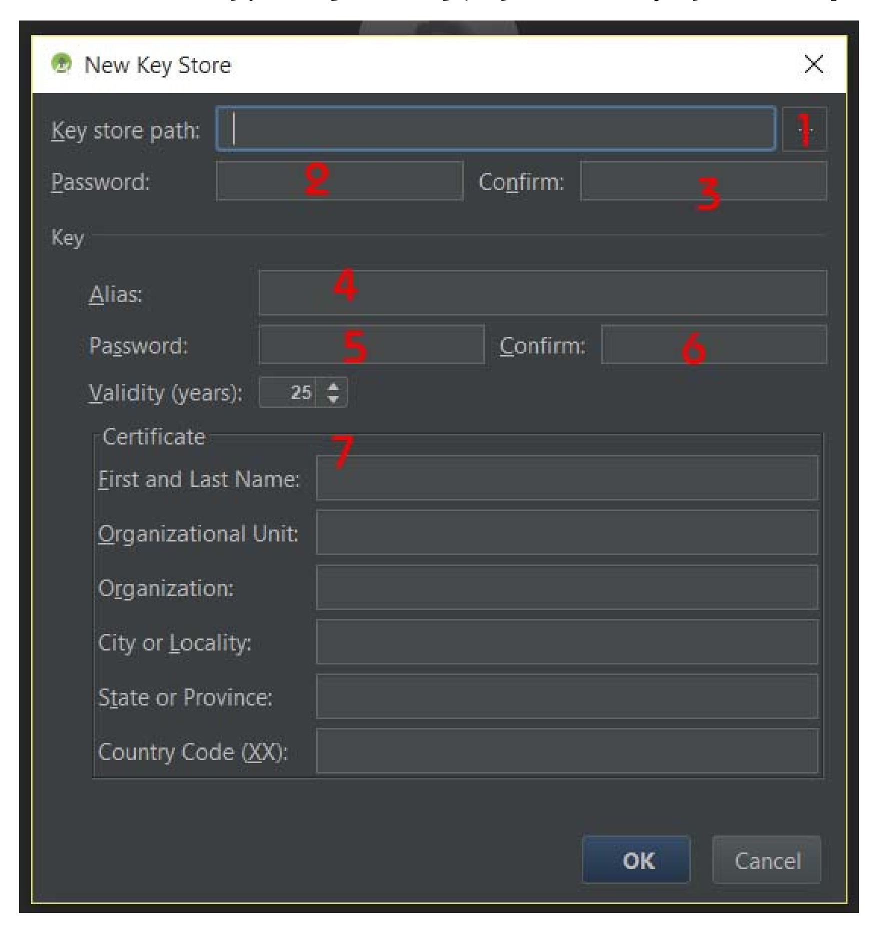
2/ لاختيار شفرة موجودة مسبقاً لدينا على الجهاز.

3/ كلمة المرور لهذه الشفرة.

4/ لاختيار التطبيق الذي تريده من هذه الشفرة.

5/ كلمة المرور لمفتاح هذا التطبيق.

سوف نضغط على إنشاء شفرة Key على الرقم واحد. سوف تظهر النافذة التالية :



حسب الأرقام الموضحة في الصورة السابقة :

1/ مكان حفظ واسم ملف الشفرة jks.

2/ كلمة المرور لهذه الشفرة.

3/ تأكيد كلمة المرور.

--- هكذا انتهينا من إنشاء شفرة المبرمج.

4/ اسم معين للتطبيق او للدلالة لهذا التطبيق. وهو المفتاح

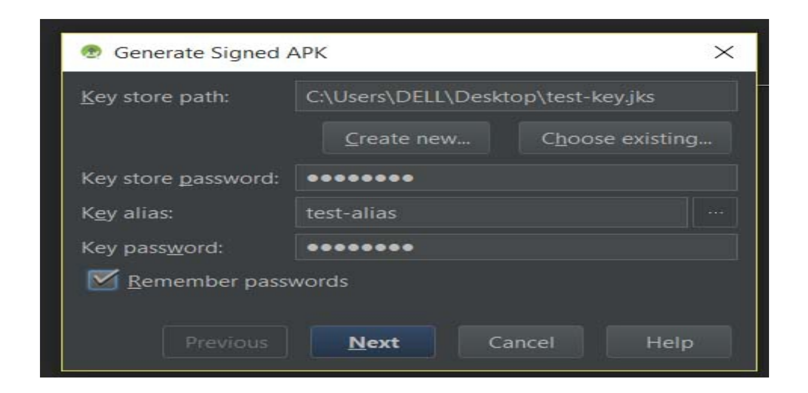
5/ كلمة المرور لمفتاح التطبيق.

6/ تاكيد كلمة المرور.

7/ بالنسبة لـ 25 سنة عمر صلاحية هذا المفتاح لا مشكلة اتركه كما هو, والمعلومات الاخرى غير مهمة فقط اكتب اسمك او إي شيء في الحقل الأول.

بعد ملأ المعلومات اضغط على **OK**.

سوف يتم الرجوع بنا إلى النافذة السابقة, ولكن المعلومات جاهزة فيها :



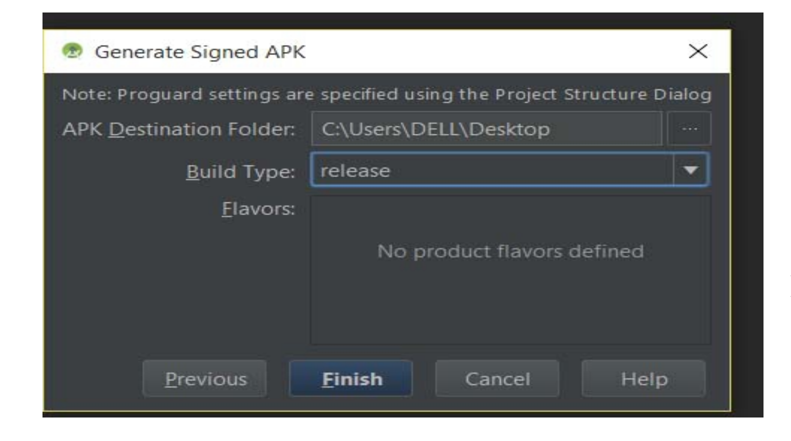
اضغط التالي واكتب كلمة المرور للشفرة :

هنا تحدد المسار الذي سوف يتم حفظ

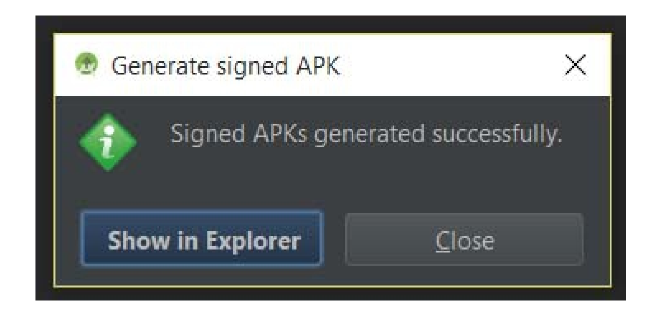
داخله ملف التطبيق APK.

وكذلك ضروري جداً ان تتأكد إن Build Type

محدد على release



الآن اضغط Finish وسوف تبدأ عملية التصدير وعند الانتهاء سوف تخرج لك نافذة :



اضغط على Show in Explorer لعرض المسار الذي تم حفظ التطبيق داخله.

في حال قمت ببرمجة تطبيقات أخرى, فقط اختر نفس ملف الشفرة وقم بإنشاء Key Alias داخل هذه الشفرة جديدة للتطبيق الجديدة.

وفي حال أردت تطوير بعض المشاريع التي تم نشرها مسبقاً, اختر نفس الشفرة و نفس المفتاح للتطبيق.

الآن ظهر لديك ملف Apk. وهو ملف التشغيل للتطبيق, وتستطيع إرساله إلى إي جهاز اندرويد لتجربته, وكذلك رفعه مباشرة على متجر Google Play . وغيرها من المتاجر مثل أمازون ون ماركت موبايل متجر سامسونج إلخ.

** الكثير منا يحتاج لمشاريع جاهزة يستفيد منها في الأكواد, ويتدرب ويتعلم عليها, وهنا سوف أذكر أهم موقع يقدم لك المشاريع الجاهزة لمشروعات الأندرويد.

https://github.com/

** عندما تريد البحث عن إي نقطة عن عملية انتقال .. مثلاً انتقل من اكتفتي إلى اكتفتي آخر, أو الانتقال من التطبيق إلى تطبيق آخر مثل مستعرض الويب, الرسائل إلخ. ابحث عن ما تريد بالإضافة إلى كلمة Intent.

** إذا كانت هناك أكواد لم تفهم معناها, فقط اضغط على زر Ctrl واضغط على الكلمة التي تريد بالماوس, حتى لو متغيرات أنت قمت بوضعها بهذه الطريقة يظهر لك إين تم استخدام هذه المتغيرات.

الخاتمة

الحمدلله رب العالمين, ما أحب أن أقوله بصراحة برنامج الأندرويد ستوديو يحتاج لشرح كتاب كامل خاص فيه, لذلك لم اشرح إلا أساسيات منه, واحتمال كبير يتم شرح استخدام البرنامج في كتاب منفصل لكن في إصدارات جديدة. كما إنني حاولت قدر الإمكان الاختصار لكثير من الأمور, فركزت على الأساس الذي نحتاجه ويوجد أمور أخرى يجب معرفتها, ولكن كلما زادت صفحات الكتاب كلما شعر القارئ بالملل لذلك سوف أبسط الأمر, لأنه يوجد أشياء مشيت فيها من الصفر تماماً, و حاولت في بعض الدروس أن اختصر, وسوف أحاول في وقت ما أن أقوم بإنشاء كتب تحتوي على مشاريع أكثر وكيفية بنائها. كما إنني كتبت في هذا الكتاب صفحات أخرى, وأخذ مني وقت طويل, ولكن عندما راجعت الكتاب حذفت الكثير من الأمور, كي أختصر عليك أخي القارئ . وفي لقاء في كتب أخرى.

كما يسعدني و**بكل أمانة إعطاء رأيك بهذا الكتاب** من خلال الرابط التالي :

http://andrody.com/?p=1393

تعليقك إن شاء الله سوف يحسن تجربتي وبالتأكيد سوف يشجعني للاهتمام أكثر في المرات القادمة. إي ملاحظة او استفسار أخبرني بها مشكوراً .

كما يسعدني تقييمك للكتاب من نفس الرابط السابق, و السلام عليكم جميعاً :)